



ITS
Institut
Teknologi
Sepuluh Nopember

PROYEK AKHIR

**ANALISA PENGIRIMAN CITRA TERKOMPRESI
JPEG DENGAN TEKNIK SPREAD SPEKTRUM
DIRECT SEQUENCE (DS-SS)**

**YUSHINTIA PRAMITARINI
7209.040.505**

Dosen Pembimbing:

**ARIES PRATIARSO, ST.MT
NIP. 196611171991031004**

**M. AGUS ZAINUDDIN, ST.MT
NIP. 197808122008011015**

**JURUSAN TEKNIK TELEKOMUNIKASI
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2011**



**ANALISA PENGIRIMAN CITRA TERKOMPRESI
JPEG DENGAN TEKNIK SPREAD SPEKTRUM
DIRECT SEQUENCE (DS-SS)**

**YUSHINTIA PRAMITARINI
7209.040.505**

Dosen Pembimbing:

**ARIES PRATIARSO, ST.MT
NIP. 196611171991031004**

**M. AGUS ZAINUDDIN, ST.MT
NIP. 197808122008011015**

**JURUSAN TEKNIK TELEKOMUNIKASI
POLITEKNIK ELEKTRONIKA NEGERI
SURABAYA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2011**

**Analisa Pengiriman Citra Terkompresi JPEG Dengan Teknik
Spread Spektrum Direct Sequence (DS-SS)**

Oleh :

YUSHINTIA PRAMITARINI
NRP. 7209.040.505

Proposal Tugas Akhir ini Diajukan Sebagai Salah Satu Syarat Untuk
Memperoleh Gelar Sarjana Sains Terapan (S.ST)
di

Politeknik Elektronika Negeri Surabaya,
Institut Teknologi Sepuluh Nopember Surabaya

Disetujui Oleh :

Tim Penguji Proyek Akhir:

Dosen Pembimbing :

1. Ir. Muhamad Milchan, MT
NIP. 196011061987011001

1. Aries Pratiarso, ST.MT
NIP. 196611171991031004

2. Ir. Yoedy Moegiharto, MT
NIP. 195805311987011002

2. M.Agus Zainuddin, ST.MT
NIP. 197808122008011015

3. Arifin, ST.MT
NIP. 196005031988031004

Mengetahui
Ketua Jurusan Teknik Telekomunikasi

Arifin, ST.MT
NIP. 196005031988031004

ABSTRAK

Saat ini penggunaan informasi citra digital digunakan secara luas dalam berbagai macam aplikasi yang membutuhkan bandwidth kanal pentransmisian dan media penyimpanan yang cukup besar. Untuk mengatasi terbatasnya media penyimpanan dan bandwidth kanal pentransmisian, maka perlu dirancang suatu sistem kompresi citra yang memiliki tingkat kemampuan kompresi yang tinggi.

Sistem kompresi citra berdasarkan algoritma JPEG Compression merupakan sistem kompresi citra yang memiliki tingkat kemampuan kompresi yang tinggi. Sesuai dengan algoritma JPEG compression, citra grayscale asli akan dikompresi pada beberapa rasio kompresi.

Dikarenakan aliran data terkompresi sangat rentan terhadap gangguan kanal. Pada proses transmisi digunakan teknik Direct Sequence Spread Spektrum (DS-SS). Hasil yang didapat setelah simulasi, berupa citra-citra rekonstruksi yang diukur dengan parameter penilaian obyektif yang berdasarkan nilai Peak Signal to Noise Ratio penilaian obyektif kualitas citra rekonstruksi yang efisien diperoleh nilai PSNR 48.23 dB dan jumlah bit per pixel 3.63 bpp.

Kata kunci: JPEG, kompresi citra, PSNR, Spread spectrum

ABSTRACT

Now the use of digital image information is widely used in various applications that require bandwidth transmission channels and storage media are quite large. To overcome the limited storage and bandwidth transmission channel, it is necessary to design an image compression system that has a high degree of compression capability.

Image compression system based on JPEG compression algorithm is an image compression system that has a high degree of compression capability. In accordance with the JPEG compression algorithm, the original grayscale images will be compressed at several compression ratios.

Due to the compressed data stream is very vulnerable to channel interference. In the process of transmission technique was used Direct Sequence Spread Spectrum (DS-SS). Results obtained after simulation, reconstruction of the images measured by an objective assessment parameters based on the value Peak Signal to Noise Ratio is an objective assessment of an efficient reconstruction image quality obtained PSNR value 48.23 dB and the number of bits per pixel 3.63 bpp.

Keywords : JPEG, image compression, PSNR, Spread spectrum

KATA PENGANTAR



Segala puji bagi Allah SWT , salam dan shalawat semoga tercurah kepada Nabi Muhammad SAW, keluarganya, para sahabatnya dan pengikutnya hingga akhir jaman.

Syukur Alhamdulillah sampai saat ini kami masih diberi kemampuan untuk dapat menyelesaikan proyek akhir yang berjudul :

“ ANALISA PENGIRIMAN CITRA TERKOMPRESI JPEG DENGAN TEKNIK SPREAD SPEKTRUM DIRECT SEQUENCE (DS-SS) ”

Pembuatan dan penyusunan proyek akhir ini diajukan sebagai syarat untuk menyelesaikan studi Diploma IV di jurusan Teknik Telekomunikasi, Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh Nopember Surabaya.

Saya menyadari bahwa apa yang saya lakukan dalam penyusunan buku proyek akhir ini masih terlalu jauh dari kata sempurna. Oleh karena itu, saya sangat mengharap kritik dan saran yang berguna dalam penyempurnaan sistem ini dimasa yang akan datang. Semoga apa yang telah saya lakukan ini dapat bermanfaat bagi pembaca.

Surabaya, Januari 2011

Penyusun

UCAPAN TERIMAKASIH

Sujud syukur dihaturkan kepada Allah SWT karena atas berkah dan rahmad-Nya sehingga penulis telah diberikan kesempatan untuk menyelesaikan proyek akhir ini, juga tak lupa penulis ucapkan terima kasih kepada :

1. Ayah dan Ibu serta Kakak tersayang, yang selalu kubanggakan kapanpun dimanapun, yang telah memberikan dukungan doa, motivasi, materi, kasih sayang dan segala-galanya yang tidak akan pernah bisa terukur nilainya. Tak ada kata yang pantas diucapkan dan budi yang dilakukan untuk membalas meskipun setetes dari lautan pengorbanan kasih sayang, materi, dukungan, serta doa yang selama ini diberikan kepadaku.
2. Bapak Dr. Ir. Dadet Pramadihanto, M.Eng selaku Direktur PENS-ITS.
3. Bapak Arifin ,ST.MT selaku ketua jurusan Teknik Telekomunikasi
4. Bapak Aries Pratiarso, ST. MT dan Bapak M. Agus Zainuddin, ST.MT selaku dosen pembimbing yang telah banyak meluangkan waktu memberikan ilmu, pengarahan, bimbingan, dan masukan-masukan kepada kami selama mengerjakan Proyek Akhir ini
5. Bapak-Ibu Dosen PENS-ITS yang tidak bisa saya sebutkan satu-persatu, terimakasih atas ilmu yang telah diberikan selama 3 semester saya belajar di PENS-ITS.
6. Teman-Teman kelas D4LJ Arinta (Buzz), Nia (Bes), Lidya, De2, Raden, Ical, Hendra ‘kucing’, Asnur dan teman-teman yang tidak bisa disebutkan satu persatu terimakasih atas solidaritasnya.
7. Si buls ku a.k.a gogo_lala33 selalu menemaniku ☺
8. Sahabatku Farah, Topik, dan Nina yang selalu menyemangati aku dalam keadaan apapun dan dimanapun. I luv U guys..
9. Semua pihak yang telah membantu dalam penyelesaian Tugas akhir ini yang tidak dapat disebutkan satu persatu, saya ucapkan terimakasih

Semoga Allah SWT senantiasa memberikan perlindungan dan memberikan balasan yang lebih di kemudian hari.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
UCAPAN TERIMA KASIH	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL.....	x

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Pembatasan Masalah	2
1.4 Tujuan	3
1.5 Metodologi	3
1.5.1 Studi Literatur	3
1.5.2 Perancangan dan Pembuatan Sistem	3
1.5.3 Analisa Kinerja Sistem dan Kesimpulan	4
1.6 Sistematika Pembahasan	4

BAB II TEORI DASAR

2.1 Umum.....	5
2.2 Pengertian Citra	5
2.2.1 Pengolahan Citra	6
2.2.2 Format Pixel 1 Bit (Citra Biner Monocrom)	7
2.2.3 Format Pixel 8 Bit (Citra Grayscale)	7
2.2.4 Format Pixel 24 Bit (Citra Warna/ True Color).....	8
2.2.5 Filtering Citra.....	9
2.3 Kompresi Citra.....	11
2.3.1 Teknik Kompresi	12
2.3.2 Algoritma Joint Photographic Expert Group (JPEG)	12
2.4 Sistem Transmisi Digital	19
2.4.1 Source Code	20
2.4.2 Spread spectrum	20
2.4.3 Direct Sequence Spread Spectrum (DS-SS)	21

2.5 Kanal BSC	27
2.6 PSNR	28

BAB III PERENCANAAN DAN PEMBUATAN SISTEM

3.1 Blog Diagram Sistem	31
3.2 Pembuatan Sistem	32
3.2.1 Penggunaan Matlab	32
3.2.2 Pembacaan Citra	35
3.2.3 Pentransformasian Citra	35
3.2.4 Pembangkitan Goldcode	37
3.2.5 Pentransmisian Kanal BSC	38
3.2.6 Proses Decoding	39
3.2.7 Perhitungan PSNR	40

BAB IV PENGUJIAN DAN ANALISA

4.1 Peralatan Uji Coba	41
4.2 Citra Awal	41
4.3 Kompresi Data	43
4.3.1 Sampling	43
4.3.2 Discrete Cosine Transform	44
4.3.3 Quantization	44
4.3.4 Entropy Coding	44
4.3.5 Decoding	46
4.4 PSNR	47
4.5 Eksperiment Citra	49
4.5.1 Citra Lena	49
4.5.2 Citra Baboon	50
4.5.3 Citra Pepper	51
4.6 Pengaruh Quality	52
4.7 Pengaruh rasio Kompresi	53
4.8 Perbandingan Citra	54

BAB V PENUTUP

5.1 Kesimpulan	57
5.2 Saran	57

DAFTAR PUSTAKA	59
----------------------	----

LAMPIRAN	59
----------------	----

RIWAYAT HIDUP PENULIS	
-----------------------	--

DAFTAR GAMBAR

Gambar 2.1	Pixel Matrik	6
Gambar 2.2	Ilustrasi Citra Biner Monocrom.....	7
Gambar 2.3	Ilustrasi Citra Grayscale.....	8
Gambar 2.4	Ilustrasi Citra True color	9
Gambar 2.5	Format Koordinat Frekuensi pada Citra.....	10
Gambar 2.6	Contoh Transformasi Fourier Citra Bergadasi Tinggi	10
Gambar 2.7	Contoh Transformasi Fourier Citra Bergadasi Rendah...	11
Gambar 2.8	Algoritma Kompresi JPEG.....	15
Gambar 2.9	Pembagian Koefisien Frekuensi DCT untuk ukuran Blok 8x8	16
Gambar 2.10	Citra dasar dengan DCT 2D	17
Gambar 2.11	Matrik Standart Kuantisasi.....	18
Gambar 2.12	Urutan Zig Zag Kuantisasi.....	19
Gambar 2.13	Sistem Transmisi Digital	19
Gambar 2.14	Building block dari sitem DSSS	21
Gambar 2.15	Proses spreading dan despreading	23
Gambar 2.16	Proses Spreading	23
Gambar 2.17	M-Sequence	24
Gambar 2.18	Konfigurasi Gold Code	25
Gambar 2.19	Ilustrasi Generator Goldcode	26
Gambar 2.20	Matrik Handmard – Walsh Code	27
Gambar 2.21	Kanal BSC	28
Gambar 3.1	Blok Diagram Sistem	31
Gambar 3.2	Blok Diagram Kompresi	32
Gambar 3.3	Diagram Alur Sistem	34
Gambar 4.1	Citra Inputan.....	42
Gambar 4.2	Pembacaan Citra	43
Gambar 4.3	Proses Sampling.....	43
Gambar 4.4	Rate Kompresi dan Ratio Kompresi	46
Gambar 4.5	Kompresi Citra Lena	47
Gambar 4.6	MSE dan PSNR Citra Lena dengan Q=50	48
Gambar 4.7	Citra Lena dengan Q=70 bpp=0.35.....	50
Gambar 4.8	Citra Baboon dengan Q=70 bpp=1.03	51
Gambar 4.9	Citra Pepper dengan Q=70 bpp=0.4	52
Gambar 4.10	Grafik hubungan Quality dan PSNR	52
Gambar 4.11	Grafik hubungan Quality dan Compression Ration	53
Gambar 4.12	Perbandingan Citra Asal dengan citra rekonstruksi	54

DAFTAR TABEL

Tabel 2-1	Macam-Macam Format Citra	12
Tabel 2-2	Feedback M-Sequence.....	25
Tabel 2-3	Gold Code Preferred Pair	27
Tabel 4-1	Spesifikasi Uji Coba	41
Tabel 4-2	Nilai PSNR	48
Tabel 4-3	Hasil Kompresi JPEG pada Citra Lena	49
Tabel 4-4	Hasil Kompresi JPEG pada Citra Baboon	50
Tabel 4-5	Hasil Kompresi JPEG pada Citra Pepper	51

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Data atau informasi saat ini tidak hanya disajikan dalam bentuk teks semata, tetapi juga dapat disajikan dalam bentuk lain misalnya gambar (images) , suara (audio), maupun video. Situs web (website) yang kita jumpai di internet biasanya dibuat semenarik mungkin dengan menyertakan gambar. Beberapa waktu lalu istilah SMS (short message service) begitu populer bagi pengguna telepon genggam (handphone) saat ini sudah bergerak ke MMS (multimedia message service) yang banyak melibatkan gambar maupun video). Keempat bentuk informasi saat ini hampir tidak dapat dipisahkan biasa dinamakan dengan multimedia.

Dari uraian di atas dapat dilihat bahwa teknologi digital saat ini mengalami perkembangan yang sangat pesat. Banyak peralatan digital yang dapat ditemui dalam kehidupan sehari-hari seperti komputer, kamera digital, dan sebagainya. Sehingga tidaklah mengherankan apabila saat ini banyak produk-produk digital yang dihasilkan. Salah satunya adalah citra digital. Citra digital sebenarnya merupakan sebuah citra yang diperoleh dari proses digitalisasi terhadap data citra analog [R Munir, 2004]. Dibandingkan dengan data teks, citra mempunyai karakteristik tersendiri, yaitu citra merupakan data yang kaya dengan informasi. Bahkan ada sebuah istilah yang cukup populer yaitu “a picture is more than a thousand words” yang berarti bahwa sebuah gambar memiliki makna lebih dari seribu kata. Maksudnya adalah bahwa sebuah gambar dapat memberikan informasi yang lebih banyak jika dibandingkan dengan informasi yang disajikan dalam bentuk kata-kata atau tulisan. Citra digital saat ini banyak digunakan dalam berbagai bidang. Mulai dari keperluan sehari-hari seperti cetak foto, pemetaan hutan, identifikasi forensik maupun sidik jari di kepolisian, rekam medis dengan menggunakan citra kedokteran (medical images) sampai pada citra satelit. Hampir semua jenis citra digital memerlukan media penyimpanan (storage) yang cukup besar. Sehingga hal ini dapat menimbulkan masalah yang cukup serius ketika citra digital disimpan dalam database dengan keterbatasan media penyimpanan yang ada. Masalah lain adalah ketika diinginkan untuk mengirimkan citra digital dengan menggunakan jalur komunikasi atau internet. Dengan ukuran yang besar maka citra digital juga memerlukan waktu pengiriman yang

lama. Sehingga diupayakan suatu teknik yang dapat mereduksi besarnya ukuran file citra digital. Salah satu teknik yang dikembangkan adalah kompresi. Banyak teknik kompresi yang dikembangkan hingga saat ini salah satunya adalah kompresi JPEG [Bandemer Bernd, 2003].

JPEG adalah algoritma kompresi secara *lossy*, bekerja dengan merubah gambar spasial dan merepresentasikan kedalam pemetaan frekuensi dimana algoritma kompresi citra tersebut mampu mencapai kompresi dengan ukuran yang kecil. *Discrete Cosine Transform (DCT)* bekerja dengan memisahkan antara informasi frekuensi yang rendah dan tinggi dari sebuah gambar. Informasi frekuensi yang tinggi akan diseleksi untuk dihilangkan yang terikat pada penganturan kualitas yang digunakan. Namun aliran data terkompresi sangat rentan terhadap gangguan kanal, meski untuk jumlah kesalahan data yang sedikit. Hal ini mensyaratkan pengkodean kanal untuk memproteksi data sebelum ditransmisikan pada kanal. Sehingga untuk proses transmisinya digunakan teknik Direct Sequence Spread Spektrum (DS-SS) tanpa sinkronisasi. Dengan teknik DS-SS digunakan kode gold dalam mengacak data yang akan dikirim.

1.2 Perumusan Masalah

Dalam pembuatan proyek akhir ini, terdapat beberapa permasalahan, antara lain sebagai berikut :

- a. Pemilihan kode yang digunakan dalam pengacakan data
- b. Data yang akan diproses menggunakan teknik kompresi JPEG
- c. Teknik pentransmisian yang dapat digunakan sebagai teknik transmisi data
- d. Parameter yang akan diukur dalam sistem pengkompresian data

1.3 Pembatasan Masalah

Dalam tugas akhir ini permasalahan di batasi dengan asumsi sebagai berikut

1. Kode yang digunakan adalah tipe gold code
2. Data citra berupa gambar yang berukuran 256 x 256 grayscale.
3. Teknik transmisi data yang digunakan ada teknik spreadspektrume direct sequence (DS-SS) tanpa sinkronisasi
4. Paramater yang akan diukur adalah PSNR (Peak Signal Noise Rasio), sedangkan BER tidak dilakukan pengukuran

dikarenakan pentransmisian spread spectrum yang digunakan tanpa sinkronisasi

1.4 Tujuan

Tujuan pembuatan proyek akhir ini diharapkan dapat mengetahui sejauh mana tingkat penurunan kualitas sebuah citra digital dan ukuran filenya jika dikenakan sebuah proses kompresi JPEG serta menganalisa kinerja teknik kompresi JPEG pada sistem transmisi *spread spectrum direct sequence (DS-SS)* tanpa sinkronisasi.

1.5. Metodologi

Dalam upaya mendapatkan data yang obyektif, maka dibutuhkan metode yang sesuai. Sehubungan dengan hal tersebut, dalam penulisan laporan proyek akhir ini menggunakan metode-metode yang diharapkan dapat dijadikan pedoman dalam penyusunan laporan proyek akhir ini. Adapun metode yang digunakan adalah sebagai berikut :

1.5.1 Studi Literatur

Dalam pembuatan proyek akhir ini harus terlebih dahulu dilakukan studi literatur tentang hal-hal yang berkaitan dengan proyek akhir ini seperti teori tentang pengertian citra, kompresi citra JPEG, serta Algoritma DCT.

1.5.2 Perancangan dan Pembuatan Sistem

Sistem yang akan dibuat nantinya berupa sebuah software simulasi kompresi citra jpeg yang di transmisikan menggunakan teknik DS-SS tanpa sinkronisasi. Sumber data yang digunakan berupa citra dengan besar pixel 256x256 (gray level 0-255). Citra yang dipilih akan di dekomposisi menjadi 8x8 block. Setelah itu masing-masing koefisien DCT akan dibagi sesuai dalam tabel standar kuantisasi dan nilainya akan dibulatkan ke integer terdekat. Langkah berikutnya, koefisien DCT yang terkuantisasi akan di urutkan ke standar zig zag. Setiap 64 blok dari koefisien DCT ditetapkan bahwa nilai terendah berada di sudut kiri atas dengan frekuensi tertinggi berada di pojok kanan bawah. DCT merubah gambar yang sudah terbagi 8x8 blok dari domain spatial ke domain

frekuensi. Kemudian bit binary yang di dapat dari proses sebelumnya akan dikodekan menggunakan kode Direct Sequence Spread Spectrume yang bertipe goldcode. Pengkodean ini merupakan kode pengacakan data yang mana agar data yang dikirim aman dari gangguan saat proses transmisi. Setelah itu data ditransmisikan dan diterima oleh receiver dan dilakukan invers dari DCT sehingga menghasilkan citra rekonstruksi dimana citra tersebut hampir sama dengan aslinya.

1.5.3 Analisa Kinerja sistem dan Kesimpulan

Setelah sebuah program simulasi selesai maka akan dilakukan analisa terhadap kinerja dari sistem tersebut.

Pertama akan dilakukan perbandingan antara ketiga jenis citra yaitu lena, baboon, dan barbara dengan nilai kuantisasi citra yang berbeda-beda. Dengan adanya perubahan kuantisasi tersebut maka akan menghasilkan nilai PSNR (Peak Signal Noise to Ratio) yang berbeda – beda.

Setelah itu akan dibuat suatu kesimpulan dari hasil analisa yang dilakukan terhadap sistem yang telah dibuat.

1.6 Sistematika Pembahasan

Buku proyek akhir ini tersusun atas beberapa bab pembahasan. Sistematika penulisan tersebut adalah sebagai berikut:

BAB 1 : Pendahuluan, menerangkan tentang latar belakang, rumusan masalah, batasan masalah, metodologi, tujuan dan sistematika penulisan.

BAB 2 : Teori Penunjang menerang tentang teori-teori yang menjadi landasan dalam pembuatan sistem.

BAB 3 : Perancangan Sistem menerangkan tentang segala sesuatu yang berkaitan dengan perancangan sistem.

BAB 4 : Pembuatan Sistem dan Pembahasan, menerangkan tentang pembuatan sistem baik interface maupun programnya serta analisa dari kinerja sistem.

BAB 5 : Penutup, merangkan tentang kesimpulan dari analisa sistem dan saran-saran berkaitan sistem pengkompresian citra.

BAB II

TEORI DASAR

2.1 Umum

Pada bab ini akan diberikan teori dasar yang melandasi permasalahan dan penyelesaiannya yang diangkat dalam Tugas Akhir ini. Teori dasar yang diberikan meliputi: pengertian tentang pengolahan citra, algoritma Joint Photographic Expert Group (JPEG), dan teknik spreadspektrum.

2.2 Pengertian Citra

Secara harfiah, citra (image) adalah gambar pada bidang dwimatra (2 dimensi). Ditinjau dari sudut pandang sistematis, citra merupakan fungsi continue dari intensitas cahaya pada bidang dwimatra (2D). Ada 2 jenis citra yaitu : citra diam dan citra bergerak. Citra diam adalah citra tunggal yang tidak bergerak, sedangkan citra bergerak adalah rangkaian citra diam yang ditampilkan secara sekuensial. Sedangkan citra digital merupakan citra yang tersusun dalam bentuk raster (grid / kisi). Setiap kotak (tile) yang terbentuk disebut pixel (picture element) dan memiliki koordinat (x,y). Sumbu x (horizontal) : kolom (column), sample sedangkan sumbu y (vertikal) : baris (row,line). Setiap pixel memiliki nilai (value atau number) yang menunjukkan intensitas keabuan pada pixel tersebut. Derajat keabuan dimana Merepresentasikan grey level atau kode warna. Kisaran nilai ditentukan oleh bit yang dipakai dan akan menunjukkan resolusi aras abu-abu (grey level resolution).

1 bit –2 warna: [0,1]

4 bit –16 warna: [0,15]

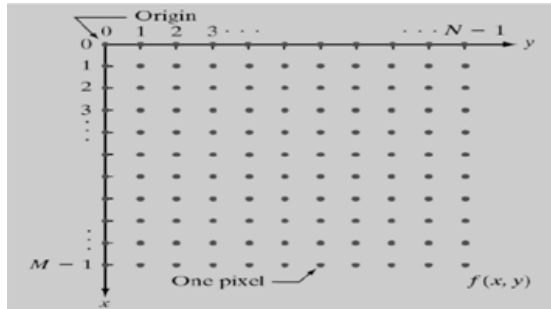
8 bit –256 warna: [0,255]

24 bit –16.777.216 warna (true color)

Kanal Merah -Red (R): [0,255]

Kanal Hijau - Green (G): [0,255]

Kanal Biru - Blue (B): [0,255]



Gambar 2.1 Pixel Matrik

Nilai Digital dan banyak bit :

M = banyak pixel per baris (panjang)

N = banyak pixel per kolom (lebar)

b = banyak / besar bit pada suatu citra

2.2.1 Pengolahan Citra

Pengolahan citra (image processing) merupakan proses untuk mengolah pixel-pixel dalam citra digital untuk tujuan tertentu. Beberapa alasan dilakukan pengolahan citra digital adalah sebagai berikut :

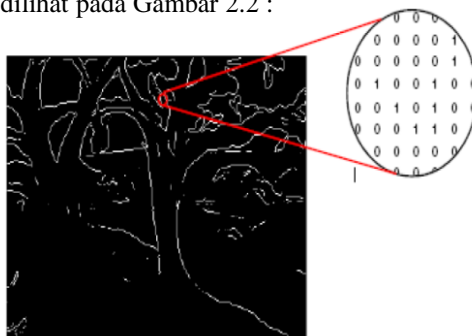
- Untuk mendapatkan citra asli dari citra yang sudah rusak karena pengaruh noise yang bercampur dengan citra asli dalam suatu proses tertentu. Proses pengolahan citra bertujuan untuk mendapatkan citra yang mendekati citra asli.
- Untuk mendapatkan citra dengan karakteristik tertentu dan cocok secara visual yang dibutuhkan dalam proses lanjut dalam pemrosesan analisis citra. Operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis sebagai berikut:
 1. Image Enhancement (Perbaikan kualitas citra)
 2. Image Restoration (Pemugaran Citra)
 3. Image Compression (Pemampatan Citra)
 4. Image Segmentation
 5. Image Analysis
 6. Image Reconstruction (Rekonstruksi Citra)

Operasi-operasi tersebut bertujuan untuk membentuk objek dari beberapa citra hasil proyeksi. Pada citra digital, dengan type bitmap type warna pada titik-titik piksel dibentuk dari sebuah data numerik. Tinggi

dan rendahnya keabuan piksel dinyatakan dalam bentuk intensitas atau derajat keabuan. Satuan lebar intensitas merupakan lebar memori (bit) citra yang disebut dengan format piksel [9].

2.2.2 Format Pixel 1 Bit (Citra Biner Monocrom)

Citra biner diperoleh melalui proses pemisahan pixel – pixel berdasarkan derajat keabuan yang dimilikinya. Pada citra biner, setiap titik bernilai 0 dan 1, masing – masing merepresentasikan warna tertentu. Nilai 0 diberikan untuk pixel yang memiliki derajat keabuan lebih kecil dari nilai batas yang ditentukan, sementara pixel yang memiliki derajat keabuan yang lebih besar dari batas akan di ubah menjadi nilai 1. Pada standard citra untuk ditampilkan di layer computer, nilai biner ini berhubungan dengan ada tidaknya cahaya yang ditembakkan oleh electron gun yang terdapat di dalam monitor computer. Angka 0 menyatakan tidak ada cahaya, dengan demikian warna yang direpresentasikan adalah hitam. Untuk angka 1, terdapat cahaya, sehingga warna yang direpresentasikan adalah putih. Standar tersebut disebut sebagai standar citra cahaya, sedangkan standar citra tinta / cat adalah berkebalikan, karena biner tersebut menyatakan ada tidaknya tinta. Setiap titik pada citra hanya membutuhkan 1 bit, sehingga setiap byte dapat menampung informasi 8 bit. Ilustrasi dari citra biner bisa dilihat pada Gambar 2.2 :

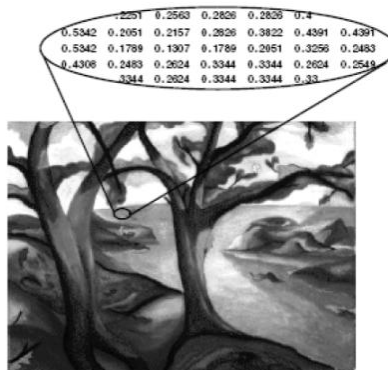


Gambar 2.2 Ilustrasi Citra Biner Monokrom

2.2.3 Format Pixel 8 Bit (Citra Gray Scale)

Citra skala keabuan memberi kemungkinan warna yang lebih banyak daripada citra biner, karena ada nilai – nilai diantara nilai minimum (biasanya = 0) dan nilai maximum. Banyaknya kemungkinan

nilai minimum dan nilai maximumnya bergantung pada jumlah bit yang digunakan. Mata manusia pada umumnya hanya mempunyai kemampuan untuk membedakan maksimal 40 tingkat skala keabuan. Untuk citra tampak / visible image dipilih skala keabuan lebih dari 40. Pada umumnya, citra skala keabuan menggunakan jumlah bit 8, sesuai dengan satuan memori computer. Contohnya untuk skala keabuan 4 bit, maka jumlah kemungkinan nilainya adalah 16, dan nilai maksimumnya adalah $2^4 - 1 = 15$. Sedangkan untuk skala keabuan 8 bit, maka jumlah kemungkinan nilainya adalah 256, dan nilai maksimumnya adalah $2^8 - 1 = 255$. Format citra ini disebut skala keabuan, karena pada umumnya warna yang dipakai adalah antara warna hitam sebagai warna minimal dan warna putih sebagai warna maksimalnya, sehingga warna diantara hitam dan putih adalah warna abu – abu. Namun pada prakteknya, warna yang dipakai tidak terbatas pada warna abu – abu. Sebagai contoh, dipilih warna minimalnya adalah putih dan warna maksimalnya adalah merah, maka semakin besar nilainya semakin besar pula intensitas warna merahnya. Format citra ini kadang disebut sebagai citra intensitas. Ilustrasi dari citra Gray Scale dapat dilihat pada Gambar 2.3:

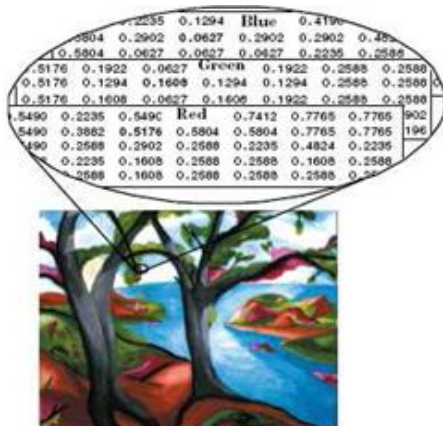


Gambar 2.3 Ilustrasi Citra Grey Scale

2.2.4 Format Pixel 24 Bit (Citra Warna/True Color)

Pada citra warna, setiap titik mempunyai warna yang paling spesifik yang merupakan kombinasi dari 3 warna dasar, yaitu merah (red), hijau (green) dan biru (blue). Ada perbedaan warna dasar untuk dasar cahaya. (misalnya display di monitor komputer) dan untuk cat (misalnya cetakan di atas kertas). Untuk cahaya, warna dasarnya adalah red green dan blue (RGB), sedangkan untuk cat warna dasarnya adalah

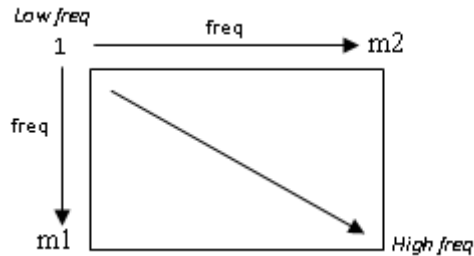
sian, magenta, kuning (cyan-magenta-yellow, CMY). Keduanya saling berkomponen. Format citra inisiering disebut sebagai citra RGB (Red-Green-Blue). Setiap warna dasar mempunyai intensitas sendiri dengan nilai maksimum 255 (8bit), misalnya warna kuning merupakan kombinasi warna merah dan hijau sehingga nilai RGB nya adalah 255 255 0. Sedangkan warna ungu muda, nilai RGB nya adalah 150 0 150, dengan demikian setiap titik pada citra warna membutuhkan data 3 byte. Jumlah kombinasi warna yang mungkin untuk format citra ini adalah 224 atau lebih dari 16 juta warna. Dengan demikian, bisa di anggap mencakup semua warna yang ada, inilah sebabnya format ini dinamakan true color. Ilustrasi dari citra ini adalah sebagai berikut :



Gambar 2.4 Ilustrasi Citra True Color

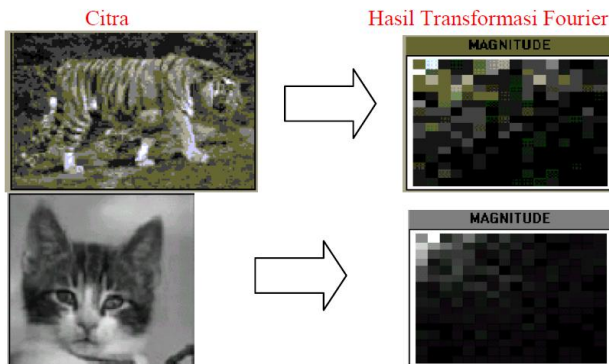
2.2.5 Filtering Citra

Filtering adalah suatu proses dimana diambil sebagian sinyal dari frekwensi tertentu, dan membuang sinyal pada frekwensi yang lain. Filtering pada citra juga menggunakan prinsip yang sama, yaitu mengambil fungsi citra pada frekwensi-frekwensi tertentu dan membuang fungsi citra pada frekwensi-frekwensi tertentu. Berdasarkan sifat transformasi fourier dari suatu citra dan format koordinat frekwensi seperti Gambar 2.5 berikut ini:



Gambar 2.5 Format Koordinat Frekuensi pada citra

Berikutnya kita perhatikan bagaimana pengaruh frekwensi rendah dan frekwensi tinggi pada citra dengan memanfaatkan hasil dari transformasi fourier. Dimana frekwensi pada citra dipengaruhi oleh gradiasi warna yang ada pada citra tersebut. Perhatikan hasil transformasi fourier dari beberapa citra berikut Gambar 2.6 berikut :

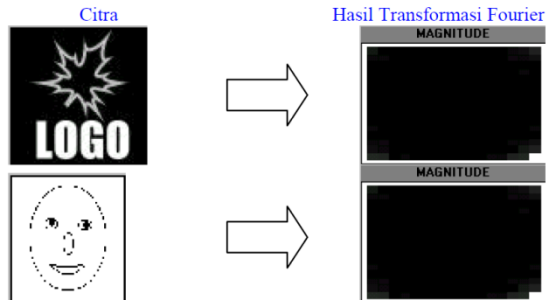


Gambar 2.6 Contoh transformasi fourier citra bergradiasi tinggi

Perhatikan bahwa warna putih (terang) pada gambar hasil transformasi fourier menunjukkan level atau nilai fungsi yang tinggi dan warna hitam (gelap) menunjukkan level atau nilai fungsi yang rendah. Berdasarkan hal ini dan format koordinat frekwensi (pada Gambar 5.) terlihat bahwa pada Gambar 2.6. nilai-nilai yang tinggi berada pada frekwensi rendah, ini menunjukkan bahwa citra dengan gradiasi (level

threshold) tinggi cenderung berada pada frekwensi rendah. Sehingga dapat disimpulkan bahwa citra dengan gradiasi tinggi berada pada frekwensi rendah.

Berikutnya dengan menggunakan citra-citra yang bergradiasi rendah seperti gambar logo data sketsa dimana nilai treshhold yang digunakan merupakan nilai-nilai yang kecil dapat dilihat pada Gambar 2.7 berikut :



Gambar 2.7 Contoh Transformasi Fourier dengan Citra gradasi Rendah

Pada Gambar 2.7. terlihat bahwa hasil transformasi fourier menunjukkan nilai fungsi hanya berada pada frekwensi tinggi. Dengan demikian dapat disimpulkan bahwa citra yang bergradiasi rendah berada pada frekwensi tinggi. Demikian pula citra biner, citra dengan threshold tertentu merupakan citra-citra yang bergradiasi rendah, dan citra-citra ini berada pada frekwensi tinggi.

2.3 Kompresi Citra

Kompresi citra merupakan suatu teknik yang digunakan untuk mengurangi biaya penyimpanan dan transmisi. Teknik-teknik yang ada yang digunakan untuk mengompresi file gambar secara luas diklasifikasikan ke dalam dua kategori, yaitu lossless dan teknik kompresi lossy.

2.3.1 Teknik Kompresi

Teknik kompresi citra terbagi menjadi 2 macam yaitu :

a. Lossy Compression

Lossy compression menyebabkan adanya perubahan data dibandingkan sebelum dilakukan proses kompresi. Sebagai gantinya lossy compression memberikan derajat kompresi lebih tinggi. Tipe ini cocok untuk kompresi file suara digital dan gambar digital. File suara dan gambar secara alamiah masih bisa digunakan walaupun tidak berada pada kondisi yang sama sebelum dilakukan kompresi.

b. Lossless Compression

Lossless Compression memiliki derajat kompresi yang lebih rendah tetapi dengan akurasi data yang terjaga antara sebelum dan sesudah proses kompresi. Kompresi ini cocok untuk basis data, dokumen atau spreadsheet. Pada lossless compression ini tidak diijinkan ada bit yang hilang dari data pada proses kompresi.

2.3.2 Algoritma Joint Photographic Expert Group (JPEG)

JPEG didirikan oleh komite Joint Photographic Expert Group yang mengeluarkan standart pada tahun 1992. JPEG menetapkan standart yaitu codec. Codec menjelaskan tentang bagaimana sebuah gambar dikompresi menjadi aliran byte dan dikompres kembali menjadi sebuah gambar serta digunakan sebagai streaming sebuah file. Algoritma kompresi JPEG merupakan yang terbaik untuk foto-foto dan lukisan pemandangan yang realistis dengan variasi warna yang halus dan senada. Berikut adalah contoh-contoh format citra :

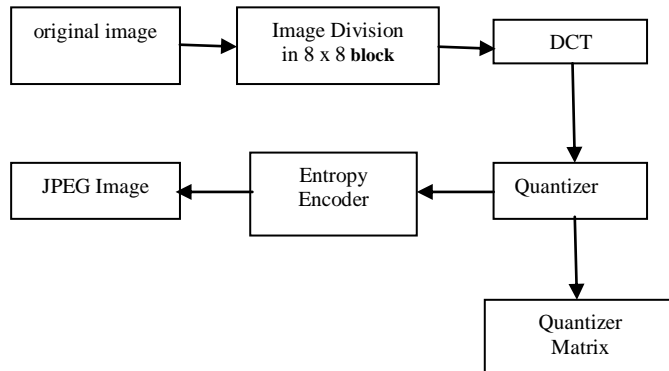
Tabel 2-1 Macam-Macam Format Citra

Ekstensi	Nama	Keterangan
Bmp	Windows Bitmap	Biasanya digunakan oleh aplikasi dan sistem operasi Microsoft Windows. Merupakan kompresi tipe lossless.
Gif	Graphics Interchange Gif	biasanya digunakan di website. Format Format gif

		<p>mendukung citra bergerak. Namun format gif hanya mendukung 255 warna tiap frame. Format gif juga Mendukung citra transparan. Format gif merupakan kompresi tipe lossy.</p>
jpg/jpeg	Joint Photographic Expert Group	<p>JPEG biasanya digunakan untuk foto pers Group atau citra di website. JPEG menggunakan kompresi tipe lossy. Kualitas JPEG 2000 bisa bervariasi tergantung setting kompresi yang digunakan. Kompresi JPEG berbasis DCT(Discrete Cosine Transform)</p>
jp2/jpg2/j2k	Joint Photographic Experts Group 2000	<p>Merupakan pengembangan dari JPEG yang berbasis transformasi wavelet. Format ini mendukung kompresi tipe lossless dan lossy. Namun, support JPEG 2000 dalam berbagai aplikasi masih kurang, disebabkan kebutuhan hardware yang tangguh dan paten</p>
PBM	Portable Bitmap Format	<p>Merupakan format citra hitam putih yang sederhana. PBM memerlukan 1 bit tiap pixel. Tidak seperti format citra lainnya, format PBM merupakan plain text yang bisa diolah dengan menggunakan pengolah text. Format PBM merupakan bagian dari PNM (Portable Pixmap File Format).</p>

PGM	Portable Graymap Format	Merupakan format citra abu-abu yang sederhana. Format PGM memerlukan 8 bit tiap pixel. PGM merupakan citra mentah dengan kompresi tipe lossless. Format PGM merupakan bagian dari PNM (Portable Pixmap File Format).
ppm	Portable Pixmap Format	Merupakan format citra berwarna yang sederhana. PPM memerlukan 24 bit tiap pixel. PPM merupakan citra mentah dengan kompresi tipe lossless. Format PPM merupakan bagian dari PNM (Portable Pixmap File Format).

Kompresi yang umum pada JPEG adalah lossy, yaitu beberapa kualitas visual akan hilang dalam proses dan tidak dapat dikembalikan. Metode kompresi lossy data dari encoding ketika diterapkan untuk input yang memiliki 24 bit per pixel (masing-masing delapan untuk merah, hijau dan biru). Metode pertama dari metode kompresi losy data adalah transformasi warna, adapun langkah pertama adalah – langkahnya adalah mengubah gambar dari RGB menjadi berbagai warna ruang (YCbCr). Y (kecerahan dan piksel), Cb dan Cr (Chrominance / biru dan merah komponen). Ruang warna YCbCr konversi memungkinkan kompresi lebih besar tanpa perceptual signifikan terhadap kualitas gambar (atau lebih besar perceptual kualitas gambar yang sama untuk kompresi). Langkah selanjutnya adalah untuk mengurangi resolusi spasial dari komponen Cb dan Cr disebut dengan *downsampling* atau *subsampling chroma*. Berikut adalah gambaran algoritma kompresi JPEG :



Gambar 2.8 Algoritma Kompresi JPEG

a. Discret Cosine Transform (DCT)

[3] DCT adalah transformasi matematika yang mengambil dan mengubah sinyal dari domain spasial ke dalam domain frekuensi. Banyak gambar digital dan skema kompresi video menggunakan blok berbasis DCT, karena algoritma ini meminimalkan jumlah data yang diperlukan untuk menciptakan gambar digital. Secara khusus, JPEG dan MPEG menggunakan DCT untuk berkonsentrasi informasi gambar dengan menghapus data spasial redundansi dalam gambar dua dimensi [4]. Dalam pengkodean JPEG standar, representasi warna dalam gambar dikonversi dari RGB ke YCbCr, maka foto tersebut membusuk dalam 8×8 blok, blok ini ditransformasi dari spasial ke domain frekuensi dengan DCT ini.

Beberapa kelebihan atau keuntungan transformasi orthogonal dibandingkan jenis transformasi lainnya adalah :

1. Koefisien- koefisien transformasi orthogonal bersifat independen sehingga lebih mudah untuk dikompresi.
2. MSE dari citra hasil rekonstruksi dapat dihitung langsung dari koefisien transformasi terkompresi tanpa perlunya melakukan *inverse transform*

Discrete Cosine Transform (DCT) digunakan sebagai transformasi standar untuk kompresi citra. Persamaan umum untuk transformasi DCT 2-dimensi dapat diperlihatkan pada persamaan 9 dan dibawah ini :

$$DCT(i, j) = \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{pixel}(x, y) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2y+1)j\pi}{2N} \right] \quad (2-1)$$

Sedangkan persamaan untuk *Inverse* transformasi DCT diperlihatkan pada persamaan 10 dibawah ini:

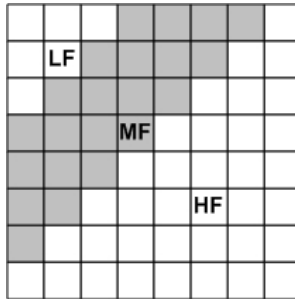
$$DCT(i, j) = \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} DCT(i, j) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2y+1)j\pi}{2N} \right] \quad (2-2)$$

dimana : $c(0) = \frac{1}{\sqrt{2}}$ untuk $i = 0$ dan $c(i) = 1$ untuk $i > 0$

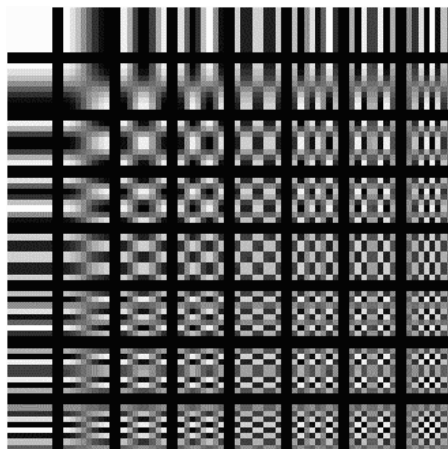
Sebelum proses transformasi, matriks transformasi C didefinisikan dengan persamaan 2-3 dibawah ini :

$$C(i, j) = \begin{cases} \frac{1}{\sqrt{N}}, i = 0 \\ \sqrt{\frac{2}{N}} \cos \frac{(2j+1)i\pi}{2N}, i > 0 \end{cases} \quad (2-3)$$

Transformasi DCT dikenal juga istilah frekuensi rendah, frekuensi menengah, dan frekuensi tinggi. Hal ini berkaitan dengan frekuensi gelombang pada fungsi basis DCT. Jika frekuensi fungsi basisnya kecil, maka koefisien yang berkorespondensi disebut koefisien frekuensi rendah.



Gambar 2.9 Pembagian Koefisien Frekuensi DCT Untuk Ukuran Blok 8 x 8



Gambar 2.10 Citra dasar dengan DCT 2D

b. Proses Kuantisasi

Tahap kuantisasi akan membuang informasi yang kurang penting yaitu informasi yang tidak berpengaruh secara signifikan terhadap visualisasi ketika sub-citra direkonstruksi. Proses kuantisasi dapat dibedakan menjadi 2 (dua) bagian yaitu:

- Operasidien koder : mengkonversi koefisien – koefisien transform menjadi *level – level kuantisasi*.
- Operasi di dekoder : mengkonversi *level –level* menjadi koefisien transform rekontruksi, biasanya disebut *inverse quantisation*.

Blok 8x8 dari koefisien DCT akan dikompresi melalui proses kuantisasi. fitur yang sangat penting dari proses jpeg berada dalam langkah ini, berbagai tingkat kompresi dan kualitas gambar dapat diperoleh melalui seleksi matriks qualization yang spesifik. Ini memungkinkan pengguna untuk menentukan tingkat kualitas berkisar antara 1 - 100, dimana angka 1 menunjukan bahwa kualitas citra yang rendah dengan tingkatan kompresi yang tinggi sedangkan angka 100 menunjukan bahwa kualitas yang dimiliki citra bernilai tinggi namun tingkatan kompresinya rendah. Dan pada akhirnya, nilai dari kualitas ato kompresi dapat disesuaikan dengan kebutuhan. Dengan tingkat quality 50, kompresi matriks keduanya bernilai tinggi dan dekompresi dari gambar pun juga bagus. Berikut adalah matrik standar kuantisasi :

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

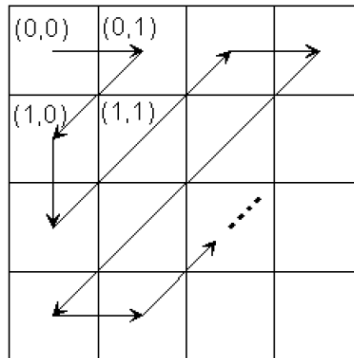
Gambar 2.11 Matrix standar kuantisasi

c. Entropy Coding

Matriks 2-D dari koefisien DCT terkuantisasi yang direpresentasikan dalam bentuk vektor satu dimensi. Setelah kuantisasi, sebagian besar koefisien frekuensi tinggi (pojok kanan bawah) adalah nol. Dalam memanfaatkan jumlah nol maka digunakan scan zig zag dari matriks. Zig zag scan memungkinkan semua koefisien DC dan AC dengan nilai yang terendah akan diproses terlebih dahulu. Koefisien DC dikodekan menggunakan diferensial encoding dan koefisien AC dikodekan menggunakan pengkodean run-length. Pengkodean Huffman digunakan untuk menyandikan kedua setelah itu .

- **Differential Encoding**
DC koefisien merupakan koefisien terbesar dalam matriks transformasi. DC koefisien memiliki variasi perlahan-lahan dari satu blok ke yang berikutnya. Hanya perbedaan nilai koefisien DC dikodekan. Jumlah bit yang diperlukan untuk pengkodeanpun berkurang.
- **Run –Length Encoding**
Koefisien AC bernilai 63. Dengan adanya zig zag scan maka terjadi nilai nol yang sangat panjang.
- **Huffman Coding**
Panjang string digit biner digantikan oleh codeword yang pendek.

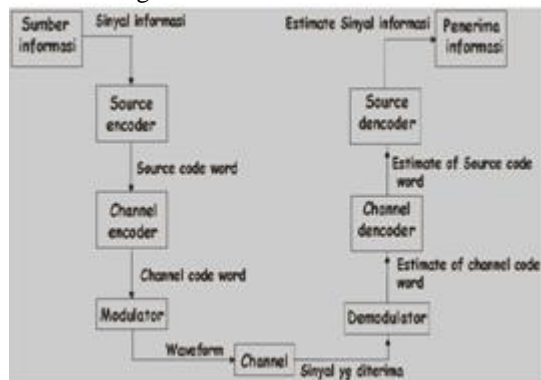
JPEG mengambil keuntungan dari ini dengan pengkodean kuantisasi zig zag urutan seperti gambar berikut :



Gambar 2.12 Urutan Zig zag Kuantisasi

2.4 Sistem Transmisi Digital

Sistem transmisi digital adalah sebuah sistem dimana data ditransmisikan menggunakan transmisi digital. Didalam system tersebut data yang sebelumnya bukan data digital diubah menjadi data digital. Sehingga data yang ditransmisikan akan berupa data yang berupa bit string. Teknik pentransmisan dapat menggunakan teknik modulator digital dan analog. Jika menggunakan system transmisi dengan radio maka kedua modulator tersebut akan digunakan. Sistem transmisi digital dapat digambarkan sebagai berikut:



Gambar 2.13 Sistem Transmisi Digital

2.4.1 Source Code

Source encoding adalah sebuah proses dimana mengubah data yang didapat dalam sebuah bahasa mesin yaitu data bit binary string. Sebaliknya source decoding adalah sebuah proses yang mana akan mengubah data dari urutan beberapa bit binary string menjadi bahasa yang diketahui oleh pembaca. Misalkan saja angka 8 dalam proses source encoding diubah menjadi 00001000, sebaliknya dalam source decoding data bit string 00001000 diubah menjadi angka 8. Dalam proses ini data yang didapat yaitu data grey level (kode warna) yang berupa data dengan type double, sehingga diperlukan pengubahan data berbentuk integer terlebih dahulu untuk membuat data tidak terdapat nilai pecahan.

2.4.2 Spread Spectrum

Lahirnya sistem komunikasi *spread spectrum* pada pertengahan tahun 1950 dilatarbelakangi oleh kebutuhan akan sistem komunikasi yang dapat mengatasi masalah interferensi, dapat menjamin kerahasiaan informasi yang dikirim dan dapat beroperasi pada tingkat S/N (*signal to noise ratio*) yang rendah atau tahan terhadap derau yang besar. Dalam sistem komunikasi sekarang ini, dimana penggunaan frekuensi sudah cukup padat sehingga interferensi dan noise dari transceiver lain cukup besar. Dalam komunikasi radio kita juga sering mendengar adanya penyadapan pembicaraan pada handphone oleh pesawat radio lain. Namun dengan sistem spread spektrum ketakutan yang dialami pada sistem komunikasi diatas akan dapat di atasi karena data yang ditransmit pada sistem spread spektrum adalah data acak yang dikenal sebagai noise. Jadi jika penerima tidak mengetahui code yang digunakan untuk melebarkan data maka penerima hanya akan menerima sinyal noise saja

Definisi dari *spread spectrum* adalah sebuah teknik transmisi dimana kode *pseudo noise*, independent dari data informasi, yang digunakan sebagai gelombang modulasi untuk “menyebarkan” energi sinyal melalui sebuah *bandwidth* jauh lebih besar dari pada *bandwidth* sinyal informasi. Pada penerima, sinyal di-“kumpulkan” menggunakan replika kode *pseudo-noise* yang telah disinkronisasikan.

Sebuah sistem spread-spectrum harus memenuhi kriteria sebagai berikut:

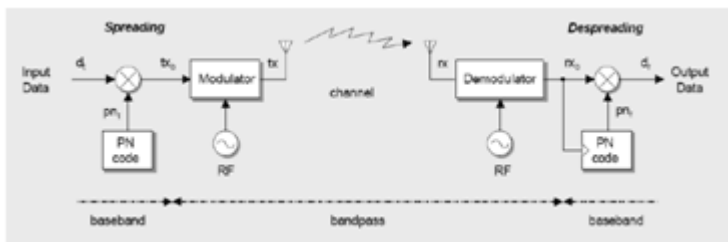
1. Sinyal yang dikirimkan menduduki bandwidth yang jauh lebihlebar daripada bandwidth minimum yang diperlukan untuk mengirimkan sinyal informasi
2. Pada pengirim terjadi proses spreading yang menebarkan sinyal informasi dengan bantuan sinyal kode yang bersifat independen terhadap informasi
3. Pada penerima terjadi proses despreading yang melibatkan korelasi antara sinyal yang diterima dan replika sinyal kode yang dibangkitkan sendiri oleh suatu generator lokal.

Ada beberapa jalan untuk menyebarkan bandwidth dari sebuah sinyal:

1. Frequency hopping. Sinyal akan berpindah-pindah frekuensi (hopping) dengan cepat dalam beberapa periode waktu tertentu.
2. Time hopping.
3. Direct sequence. Data digital di kodekan oleh bit-bit yang mempunyai kecepatan lebih tinggi dari kecepatan data.kode dibangkitkan secara random, pada sisi penerima dengan kode yang sama sinyal data diperoleh kembali.

2.4.3 Direct Sequence Spread Spectrum (DS-SS)

Sebuah *pseudo noise sequence* pnt dibuat pada modulator, yang digunakan sebagai konjungsi dengan sebuah modulasi PSK M-ary untuk menggeser fase dari PSK secara psudorandom pada *chipping rate* $R_c (=1/T_c)$, yaitu sebuah frekuensi yang berupa perkalian integer dari $R_s (=1/T_s)$. Bandwidth yang ditransmisikan ditentukan oleh chip rate dan baseband filtering. Modulasi PSK memerlukan demodulasi yang koheren.



Gambar 2.14 Building block dari sistem DSSS

- Spreading:

Pada transmitter, binary data dt (untuk BPSK, I dan Q untuk QPSK) adalah secara langsung dikalikan dengan PN sequence pnt yang terpisah dari baseband yang binary data, untuk memproduksi sinyal baseband yang ditransmisikan txb .

$$txb = dt \cdot pnt \quad \dots\dots\dots (2-4)$$

Efek dari perkalian dt dengan PN sequence adalah untuk menyebarkan baseband bandwidth Rb dari dt ke baseband bandwidth Rc .

- Despreading

Sinyal Spread Spectrum tidak bias dideteksi dengan penerima narrowband konvensional. Pada receiver, sinyal baseband rxb yang diterima dikalikan dengan PN sequence pnt .

Jika $pnr = pnt$ dan disinkronisasi ke PN sequence pada data yang diterima, kemudian binary data yang dipulihkan sd produksi pada akibat perkalian dari sinyal spread spectrum rxb dengan PN sequence pnt digunakan pada transmitter adalah untuk despread bandwidth rxb ke Rs . Dan pada saat $pnr \neq pnt$, kemudian tidak terjadi despread. Sinyal dr memiliki spread spectrum. Penerima tidak mengetahui PN sequence dari transmitter sehingga tidak bisa memproduksi kembali data yang telah dikirim.

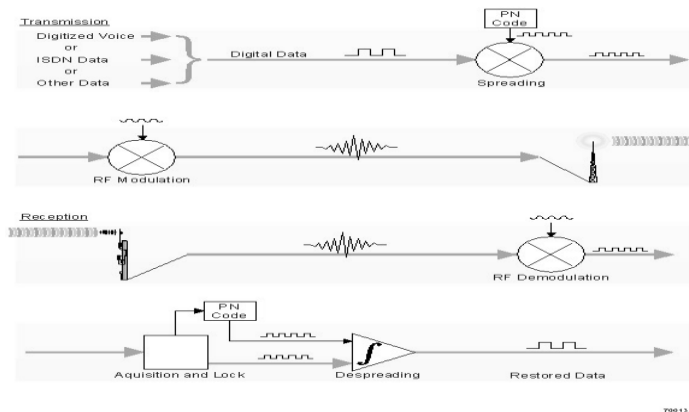
Dalam sistem direct sequence spread spectrum, sebuah sinyal ditransmisikan melalui beberapa tahap:

1. Sebuah kode random (pseudo random code) atau PN code dibangkitkan. Kode yang dibangkitkan selalu berbeda untuk tiap pengguna satu dengan yang lain.
2. Data informasi di modulasi oleh pseudo random code (proses spreading).
3. Hasil spreading kemudian dimodulasi lagi oleh frekuensi carrier.
4. Hasil modulasi tersebut akan dikuatkan dayanya oleh amplifier dan kemudian akan ditransmisikan melalui udara

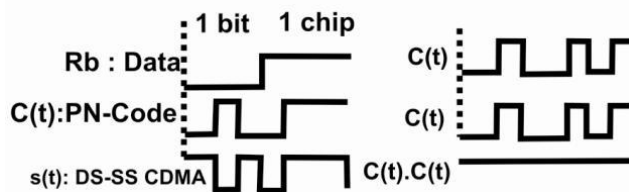
Pada sisi penerima akan dilakukan langkah-langkah sebagai berikut:

1. Sinyal akan diterima dan dikuatkan kembali oleh amplifier
2. Sinyal ayng diterima akan didemodulasi oleh lokal carrier untuk mendapatkan sinyal tersebar (spreading signal).
3. Pseudo random akan dibangkitkan. Kode random yang dibangkitkan harus sama dengan yang digunakan di sisi pengirim.
4. Setelah itu dilakukan proses sinkronisasi untuk mendapatkan waktu yang tepat kapan proses *despreading* dimulai.

5. Kemudian akan dilakukan proses *despreding* untuk mendapatkan data informasi kembali.



Gambar 2.15 Proses *spreading* dan *dispreding*



Gambar 2.16 Proses Spreading

Gambar diatas merupakan ilustrasi proses *spreading* sinyal. Sinyal data informasi di XNOR dengan sebuah PN code. Setelah proses ini akan didapatkan sinyal tersebar (spreading signal). Pada sisi penerima untuk mendapatkan data kembali sinyal tersebar (spreading signal) di XNOR kembali dengan PN code yang sama dengan yang digunakan pada pengirim.

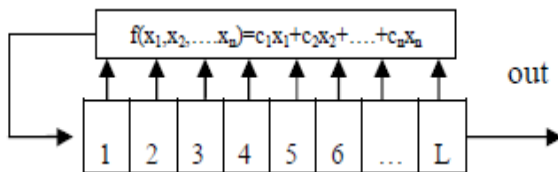
a. Pembangkit Pseudo Noise (PN)

Dalam CDMA kanal komunikasi tidak dibagi-bagi berdasarkan waktu atau frekuensi. Pemisahan atau pembagian kanal didasarkan pada kode-kode tertentu yang dibangkitkan secara acak semu (tidak benar-benar acak, melainkan mempunyai pola tertentu). Dan di sisi penerima kode yang sama seperti yang digunakan pada pengirim digunakan untuk mendapatkan kembali sinyal data informasi. Untuk itu kode-kode random ini harus mempunyai sifat-sifat sebagai berikut:

1. Harus berbeda antara satu dengan yang lain, tetapi yang digunakan pada sisi pengirim dan pada sisi penerima harus sama.
2. Harus acak, tetapi memiliki pola tertentu.
3. Cross korelasi di antara dua kode yang berbeda harus kecil
4. Kode harus mempunyai periode yang panjang.

• Kode Pseudonoise Max-Length

M-sequence adalah salah satu kode penebar (PN code) yang sering digunakan dalam sistem cdma. M-sequence dapat dibangkitkan dari sebuah Linear Feedback Shift Register. Pada metode ini digunakan beberapa shift register yang tersusun dan umpan balik dengan pola tertentu. Sebuah generator shift register sederhana (SSRG) menerima sinyal feedback yang dikembalikan kepada single inputan. Sebuah SSRG bersifat linier bila fungsi feedbacknya dapat di XOR. Secara blok diagram dapat dilihat sbb:



Gambar 2.17 M-Sequence

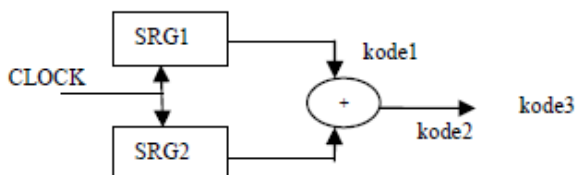
Berikut ini adalah Tabel 2.1 feedback m-sequence yang dihasilkan dari linier SSRG:

Tabel 2-2 Feedback M-Sequence

L	$N_c=2^L-1$	Feedback Taps for m-sequence	#m-sequence
2	3	[2,1]	2
3	7	[3,1]	2
4	15	[4,1]	2
5	31	[5,3] [5,4,3,2] [5,4,2,1]	6
6	63	[6,1] [6,5,2,1] [6,5,3,2]	6
7	127	[7,1] [7,3] [7,3,2,1] [7,4,3,2] [7,6,4,2] [7,6,3,1] [7,6,5,2] [7,6,5,4,2,1] [7,5,4,3,2,1]	18
8	255	[8,4,3,2] [8,6,5,3] [8,6,5,2] [8,5,3,1] [8,6,5,1] [8,7,6,1] [8,7,6,5,2,1] [8,6,4,3,2,1]	16
9	511	[9,4] [9,6,4,3] [9,8,5,4] [9,8,4,1] [9,5,3,2] [9,8,6,5] [9,8,7,2] [9,6,5,4,2,1] [9,7,6,4,3,1] [9,8,7,6,5,3]	48
10	1023	[10,3] [10,8,3,2] [10,4,3,1] [10,8,5,1] [10,8,5,4] [10,9,4,1] [10,8,4,3] [10,5,3,2] [10,5,2,1] [10,9,4,2] [10,6,5,3,2,1] [10,9,8,6,3,2] [10,9,7,6,4,1] [10,7,6,4,2,1] [10,9,8,7,6,5,4,3] [10,8,7,6,5,4,3,1]	60
11	2047	[11,2] [11,8,5,2] [11,7,3,2] [11,5,3,2] [11,10,3,2] [11,6,5,1] [11,5,3,1] [11,9,4,1] [11,8,6,2] [11,9,8,3] [11,10,9,8,3,1]	176

- **Gold Code**

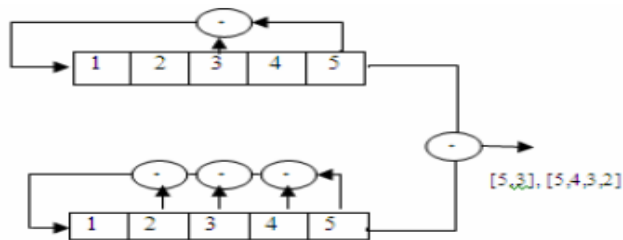
Gold codes didapatkan dari proses XOR (modulo 2 adding) dua buah maksimum sequence dengan panjang yang sama (kode faktor). Seperti gambar konfigurasi generator gold code berikut:

**Gambar 2.18** Konfigurasi Gold Code

Dengan menggunakan Panjang maksimal (maximal length) yang dapat dihasilkan adalah $2^n + 1$ (M-sequence itu sendiri dan $2^n - 1$ dari kombinasi yang dihasilkan dari pergeseran posisi chip) atau bisa disebut sebagai periode sequence. Apabila dipilih M-sequence khusus atau yang disebut preferd pair, maka Gold code yang dibangkitkan mempunyai 3 nilai cross correlation, yaitu -1 , $-2^{[(n+2)/2]} - 1$, $2^{[(n+2)/2]} + 1$

$$\text{Dimana (n)} \quad \begin{cases} 2^{[(n+2)/2]} + 1 & \text{untuk n odd ;} \\ 2^{[(n+2)/2]} - 1 & \text{untuk n even ;} \end{cases}$$

Ilustrasi dari generator goldcode dapat diperlihatkan dengan gambar sebagai berikut:



Gambar 2.19 Ilustrasi Generator Gold Code

Setiap perpindahan posisi fase antara dua generator akan menyebabkan dibangkitkannya kode-kode baru. Generator goldcode 31 chips ini akan menghasilkan deretan kode sebanyak 33 chip. Kode ini merupakan komposisi yaitu 2 kode berasal dari masing-masing generator dan 31 kode merupakan hasil dari pergeseran komponen yang ada pada sequence. Pada sequence bila dilakukan pergeseran tap maka output bit ke 32 merupakan pengulangan bit awal. Berikut ini akan diberikan contoh pembangkitan goldcode dari 2 squence yang mempunyai deret kode generator sebanyak 5 stage ($2n-1$).

Contoh:

Sequence1 [5,3] =[0 0 1 0 1]

Sequence2 [5,4,3,2] =[0 1 1 1 1]

Hasil pergeseran sequence:

Sq1 = 0 1 0 0 0 1 0 0 1 0 1 1 0 0 1 1 1 1 0 0 0 1 1 0 1 1 1 0 1

Sq2 = 1 1 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1

Gold = 1 0 1 0 0 1 1 0 1 0 0 1 1 0 1 1 0 0 1 0 0 1 0 1 1 1 0 0 1 1 0

Tabel 2-3 Gold Code Preferred Pair

L	$N_c=2^L-1$	preferred pairs of m-sequences	3-value crosscorrelations			bound
5	31	[5,3] [5,4,3,2]	7	-1	-9	-29%
6	63	[6,1] [6,5,2,1]	15	-1	-17	-27%
7	127	[7,3] [7,3,2,1] [7,3,2,1] [7,5,4,3,2,1]	15	-1	-17	-13%
8*	255	[8,7,6,5,2,1] [8,7,6,1]	31	-1	-17	+12%
9	511	[9,4] [9,6,4,3] [9,6,4,3] [9,8,4,1]	31	-1	-33	-6%
10	1023	[10,9,8,7,6,5,4,3] [10,9,7,6,4,1] [10,8,7,6,5,4,3,1] [10,9,7,6,4,1] [10,8,5,1] [10,7,6,4,2,1]	63	-1	-65	-6%
11	2047	[11,2] [11,8,5,2] [11,8,5,2] [11,10,3,2]	63	-1	-65	-3%

- **Hadamard-Walsh Code**

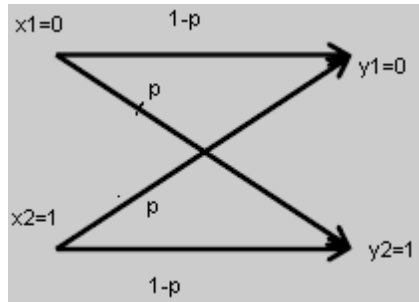
Kode ini dibangkitkan dengan menggunakan proses iteratif dari matriks Hadamard dengan sejumlah $N=2_n$ dengan panjang yang sama.

$$H_N = \begin{pmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{pmatrix}, \text{ dengan } H_0 = [1]$$

Gambar 2.20 Matriks Handamard-Walsh Code

2.5 Kanal BSC

Kanal biner simetris merupakan model kanal yang digunakan sebagai dasar dari teori informasi klasik. Mempunyai Input dan output yang berupa bilangan biner, yaitu input ($x_1=0, x_2=1$), dan output ($y_1=0, y_2=1$). Kanal ini simetris karena mempunyai probabilitas yang diterima “1” jika “0” dan sama dengan probabilitas diterima 0 jika 1 dikirim.



Gambar 2.21 Kanal BSC

Dalam penurunan dan penjabaran kanal ini, kapasitas kanal merupakan sebagai sebuah fungsi dari probabilitas transisi. Informasi Mutualnya dapat dinyatakan sebagai berikut:

$$I(X|Y) = \sum_{k=1}^K \sum_{j=1}^J p(y_k|x_j) p(x_j) \log_2 \left(\frac{p(y_k|x_j)}{\sum_{j=1}^J p(y_k|x_j) p(x_j)} \right) \dots\dots\dots (2-5)$$

2.6 PSNR (Peak Signal Noise to Ratio)

Penilaian kualitas citra diukur berdasarkan perubahan nilai tiap pixel pada citra terkompresi dan citra asal dengan menggunakan persamaan matematis tertentu. Dalam kompresi citra terdapat suatu standart pengukuran error kompresi , yaitu:

- Mean Square Error (MSE) yaitu sigma dari jumlah error antara citra hasil kompresi dan citra asal.

$$MSE = \frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left[f(i, j) - f'(i, j) \right]^2 \dots\dots\dots (2-6)$$

Dimana :

MSE = Nilai *Mean Square Error* dari citra tersebut

m = panjang citra tersebut (dalam piksel)

n = lebar citra tersebut (dalam piksel)

f(i,j)=nilai pixel citra asal

f'(i,j)=nilai pixel citra terekonstruksi.

- b. PSNR merupakan parameter yang digunakan untuk mengetahui kualitas (validasi) citra hasil kompresi. Nilai PSNR dihitung dari kuadrat nilai maksimum sinyal dibagi dengan MSE. Apabila diinginkan PSNR dalam desibel, maka nilai PSNR akan menjadi sebagai berikut :

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \dots\dots\dots (2-7)$$

Dimana :

PSNR = nilai PSNR citra (dalam dB)

MSE = nilai MSE.

=Halaman ini sengaja dikosongkan=

BAB III

PERANCANGAN DAN PEMBUATAN SISTEM

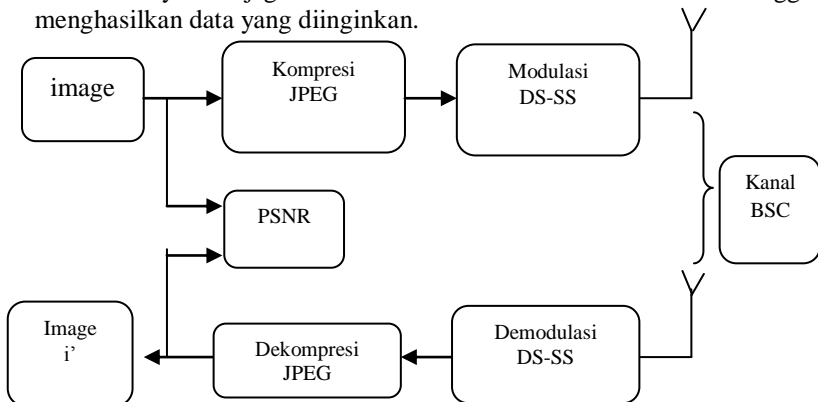
Pada bab ini akan di bahas mengenai bagaimana pembuatan dari sistem ini beserta langkah-langkahnya dan bagaimana hasil dari pembuatan sistem tersebut.

Pada sistem ini akan dibuat satu persatu mulai yang pertama yaitu membangkitkan data, pembangkitan pseudonoise generator, modulator, pembangkitan kanal awgn, demodulator. Seluruh sistem ini menggunakan perangkat lunak Matlab versi 7.0.

Setelah sistem secara keseluruhan sudah terbuat maka dilakukan tahap berikutnya yaitu tahap pengujian sistem. Dalam pengujian sistem, data-data yang dihasilkan akan dianalisa sehingga tujuan dari proyek akhir ini terpenuhi yaitu citra rekontruksi yang dihasilkan mendekati citra asal.

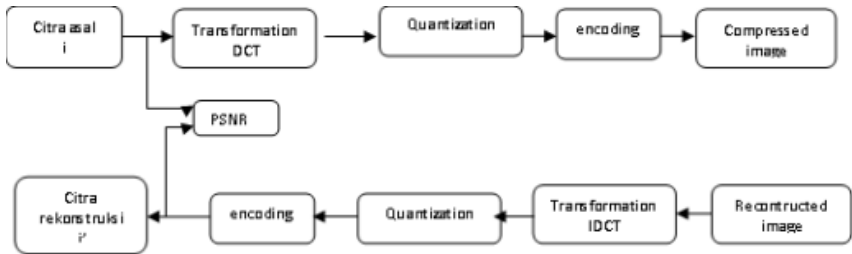
3.1 Blok Diagram Sistem

Pada Proyek Akhir ini, pertama file citra akan diubah menjadi data digital, sehingga akan diubah menjadi bit-bit biner, kemudian data digital tersebut akan dikompresi dengan metode JPEG, Setelah itu akan dikirim dengan metode Spread Spectrum direct sequences. Dengan metode spreadspektrum, data yang akan dikirim diacak terlebih dahulu agar tidak dapat dibajak oleh orang lain. Setelah data diacak agar kerahasiannya terjaga barulah didemodulasi kembali sehingga menghasilkan data yang diinginkan.



Gambar 3.1 Blok diagram sistem

Dari blok sistem diatas dapat dibagi menjadi 2 bagian yaitu pada sistem kompresi dan sistem transmisi. Berikut adalah blok sistem dari keduanya :



Gambar 3.2 Blok Diagram Kompresi JPEG

Pada gambar 3.2 disisi transmitter citra asal akan ditransformasi menggunakan Discrete Cosine Transform (DCT) kemudian dilakukan kuantisasi dan dikodekan menjadi bit-bit binary. Pada receiver didapatkan citra yang terekonstruksi kemudian dilakukan pentransformasian kembali menggunakan metode yang sama yaitu DCT baru dilakukan kuantisasi kembali sehingga menghasilkan citra yang terkonstruksi.

3.2 Pembuatan Sistem

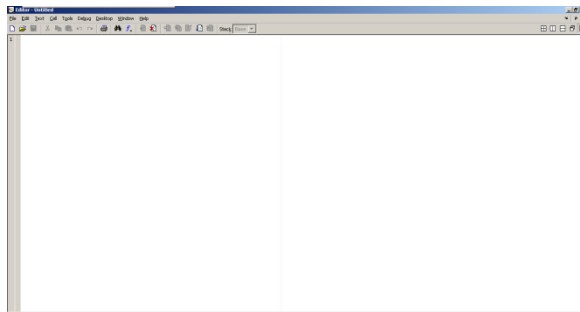
3.2.1 Penggunaan Matlab

Matlab adalah program komputer dan sekaligus bahasa pemrograman komputer generasi ke-empat yang dikembangkan oleh Grup MathWorks untuk keperluan bidang komputasi numeris dan manipulasi matriks. Untuk memulai menggunakan aplikasi ini, lakukan langkah-langkah berikut :

1. Klik menu **File** lalu pilih **New**, lalu pilih **M-File**

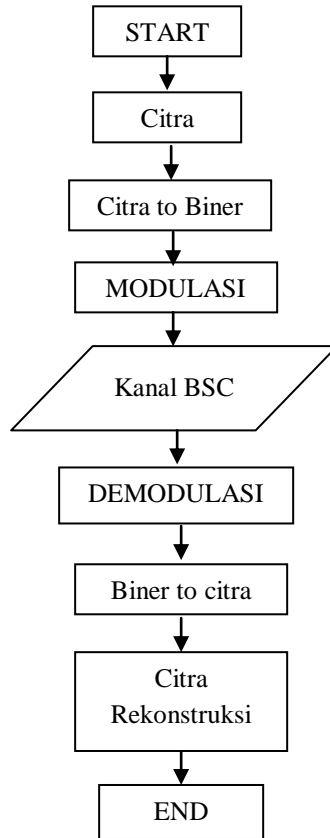


2. Lalu akan **muncul text editor**, ketikkan coding/listing di sini



3. Lalu setelah mengetikkan coding/listing, kemudian **save as**.
4. Namai filenya, lalu save
5. Jika file yang anda buat menggunakan gambar atau objek lain, simpanlah gambar tersebut ke **folder work**. Folder work berada di dalam folder utama Matlab, ingatlah lokasi Anda menginstall Matlab pada komputer Anda. Setelah folder work ditemukan simpanlah file gambar atau file lainnya yang mendukung coding/listing Anda, kemudian refresh
6. Kembali ke matlab bagian editornya, untuk mengkompaile atau menjalankan program dan menghasilkan output tekanlah **F5** pada keyboard

Pembuatan sistem seluruhnya menggunakan perangkat lunak matlab versi 7.0. Dan sistem yang akan dibuat sesuai dengan flowchart berikut ini :



Gambar 3.3 Diagram Alur Sistem

Pembuatan sistem ini berawal dari pemilihan citra yang akan di kirim. Citra yang dipilih yaitu citra yang bertipe grayscale dengan ukuran 256x256. Setelah menentukan citra mana yang akan diproses, barulah dilakukan perubahan citra kedalam bentuk binary. Bit-bit binary

yang didapat akan diolah dengan menggunakan modulasi Direct Sequence – Spread Spectrume (DS-SS). Didalam modulator, dilakukan terlebih dahulu pembangkitan data yang kemudian akan diolah dengan bit-bit dari citra asal. Dalam membangkitkan PN code digunakan gold kode karena kode ini merupakan kode yang paling bagus dalam membangkitkan PN-code.

Sedangkan pada sisi receiver data yang telah mengalami pengacakan akan diterima oleh penerima dan langsung dilakukan despreading sehingga menghasilkan bit-bit awal kembali. Setelah bit-bit itu muncul baru dikembalikan ke bentuk citra dan menghasilkan citra terekonstruksi.

3.2.2 Pembacaan Citra

Pertama kali yang dilakukan adalah membaca citra terlebih dahulu. Disini citra yang dipilih bertipe *.bmp, grayscale, serta berukuran 256x256. Berikut program yang digunakan untuk membaca citra yang diinginkan :

```
%----- Baca Citra -----
I = imread('came.bmp');
f = I;
subplot 121
imshow(I),title ('original')
```

3.2.3 Pentransformasian Citra

Dalam mentransformasi citra harus dilakukan terlebih dahulu perubahan tipe citra yang awalnya bertipe integer ke dalam variabel double. Dengan berubahnya tipe tersebut maka barulah citra dapat diproses dengan baik. Untuk merubah citra menjadi bentuk bit-bit binary maka dilakukan pentransformasian terlebih dahulu. Transformasi yang digunakan adalah DCT (Discrete Cosine Transform). Berikut program pentransformasian citra dengan DCT :

```
%----- DCT -----
f = double(f);
f = f - 128;
```

```

[mf,nf]=size(f);
mb=mf/8; nb=nf/8;

Ff = blkproc(f,[8 8],'dct');
Ff = blkproc(Ff',[8 8],'dct');
Ff = round(Ff');

Q = zeros(8);
for i = 1:8,
    for j = 1:8,
        Q(i,j) = 50;
    end
end

Fq=round(blkproc(Ff,[8 8],'divq',Q));

if mb*nb > 1,
    fdc=reshape(Fq(1:8:mf,1:8:nf)',mb*nb,1);
    fdpcm=dpcm(fdc,1);
else
    fdpcm=Fq(1,1);
end

dccof=[];
for i=1:mb*nb,
    dccof=[dccof jdcenc(fdpcm(i))];
end

z=[1    2    6    7   15   16   28   29
    3    5    8   14   17   27   30   43
    4    9   13   18   26   31   42   44
   10   12   19   25   32   41   45   54
   11   20   24   33   40   46   53   55
   21   23   34   39   47   52   56   61
   22   35   38   48   51   57   60   62
   36   37   49   50   58   59   63   64];

acseq=[];
for i=1:mb
    for j=1:nb
        tmp(z)=Fq(8*(i-1)+1:8*i,8*(j-1)+1:8*j);
    end
end

```

```

    % tmp is 1 by 64
    eobi=max(find(tmp~=0)); %end of block
index
                                % eob is labelled with 999
    acseq=[acseq tmp(2:eobi) 999];
end
end
accf=jacenc(acseq);

```

dimana citra yang sudah bervariasi double diproses dengan menggunakan teknik DCT. Pixel-pixel dari citra grayscale 0-255 namun DCT dirancang agar dapat bekerja pada pixel antara -128-127. Selanjutnya data dari citra akan didekomposisi menjadi 8x8 blok. blok ini berubah dari spasial ke domain frekuensi dengan DCT ini. Kemudian masing-masing koefisien DCT dibagi sesuai dengan tabel kuantisasi dan dibulatkan ke dalam integer terdekat. Di setiap 64 blok DCT ditetapkan bahwa nilai terendah berada disudut kiri atas dengan frekuensi tertinggi di posisi pojok kanan bawah.

3.2.4 Pembangkitan Gold Code

Gold code adalah salah satu non orthogonal code yang merupakan turunan dari M-Sequence. Gold code disusun oleh dua buah M-Sequence yang masing-masing outputnya ditambahkan (adder modulo 2). Kedua output dari M-Sequence ditambahkan (XOR) secara chip per chip menggunakan puls-pulsa clock yang sinkron. Berikut ini adalah listing program untuk membangkitkan gold code :

```

%gold_sequence
Xa=[1 1 0 0 1 1 1 1];
Xb=[1 0 0 0 0 1 1 1];
for w=1:255

    Xa=[mod(Xa(8)+Xa(7)+Xa(6)+Xa(5)+Xa(2)+Xa(1),2)
        Xa(1:7)];
    seq1(w)=Xa(8);
    Xb=[mod(Xb(8)+Xb(7)+Xb(6)+Xb(1),2)
        Xb(1:7)];
    seq2(w)=Xb(8);
    goldcode(w)=mod(Xa(8)+Xb(8),2);
end

```



```

end
for m=2:255
    goldcode(m,:)=wshift('1D',goldcode(m-
1,:),1);
end
gld=reshape(goldcode,1,[]);

```

Pada program diatas terlihat yang pertama dilakukan adalah membangkitkan dua buah M-sequence dengan 8 shift register. Squence 1= [8 7 6 5 2 1] dengan sequence2 = [8 7 6 1].

3.2.5 Pentransmisian dengan Kanal BSC

Hasil dari proses sebelumnya akan ditransmisikan lewat sebuah kanal yang mana dalam tugas akhir ini digunakan kanal BSC (Binary Symetric Channel). Berikut program yang digunakan :

```

%----kanal BSC-----
y1=[accof(1:end)];
temp=10^(-3);
for v=1:length(y1)
    r=rand(1);
    if(r<temp);
        y1(v)=(accof(v));
    end
end

y2=[dccof(1:end)];
temp=10^(-3);
for s=1:length(y2)
    r=rand(1);
    if (r<temp);
        y2(s)=(dccof(s));
    end
end
[N,BER]=biterr(accof,y1);
[N,BER]=biterr(dccof,y2);

```

3.2.6 Proses Decoding

Setelah proses pengacakan data dan pengiriman selesai baru dilakukan pengkodean kembali sehingga menghasilkan output yang berupa image yang terlihat hampir sama namun bila diamati sedetail mungkin terlihat adanya perbedaan antara image original dengan image yang sudah mengalami rekonstruksi. Berikut adalah proses pengkodean kembali yang mana menggunakan proses DCT untuk mendapatkan gambar yang sudah terekonstruksi :

```
%----- Proses Decoding -----

acarr=jacdec(y1);
dcarr=jdcdec(y2);

z=z(:);

Eob=find(acarr==999);
kk=1;ind1=1;n=1;
for ii=1:mb
    for jj=1:nb
        ac=acarr(ind1:Eob(n)-1);
        ind1=Eob(n)+1;
        n=n+1;
        ri(8*(ii-1)+1:8*ii,8*(jj-
1)+1:8*jj)=dezz([dcarr(kk) ac zeros(1,63-
length(ac))]);
        kk=kk+1;
    end
end

iFq=round(blkproc(ri,[8 8],'idivq',Q));
iFf=blkproc(iFq,[8 8],'idct2');
iFf=round(iFf+128);
subplot 122
imshow(mat2gray(iFf)),title('Citra
Rekonstruksi JPEG')
```

3.2.7 Perhitungan PSNR

Proses rekonstruksi gambar sudah selesai baru kita dapat menghitung nilai PSNR dari gambar yang diproses. Berikut adalah script dalam mendapatkan nilai PSNR :

```
% perhitungan MSE , SNR
f = f + 128;
MSE = sum(sum((f-iFf).^2))/(256^2)
PSNR = 10*log10(255^2/MSE)
```

BAB IV

PENGUJIAN HASIL DAN ANALISA

Bab ini berisi tentang hasil-hasil dari program yang telah dibuat. Dari program-program yang telah dibuat akan dilakukan pengujian. Kemudian dari data-data yang diperoleh akan dilakukan suatu analisa tentang kinerja sistem.

4.1 Peralatan Uji Coba

Pada sub bab ini menjelaskan mengenai perangkat apa saja yang digunakan dalam tugas akhir ini. Perangkat yang digunakan meliputi perangkat lunak dan perangkat keras. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam uji coba seperti pada tabel 4.1 :

Table 4.1 Spesifikasi Perangkat Uji coba

Perangkat Keras	Processor Intel(R) Core(TM) i3
	Memori 1,87GB
Perangkat Lunak	OS : Windows 7
	Perangkat Lunak pembangun : Matlab 7.1

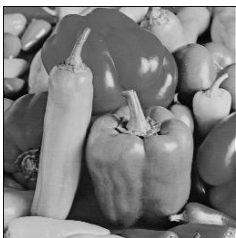
4.2 Citra Awal

Pada tugas akhir ini dilakukan simulasi pengkompresian 3 citra standar yang diantaranya Lena , Pepper, dan Baboon. Ketiga citra tersebut memiliki ukuran pixel yang sama yaitu 256x256 dengan resolusi 8 bit (gray level 0-255). Masing – masing citra dipilih karena ketiga nya memiliki kriteria yang berbeda-beda yaitu berdasarkan aktivitas dari frekuensi tiap-tiap citra. Berikut adalah 3 citra yang akan digunakan dalam tugas akhir ini :



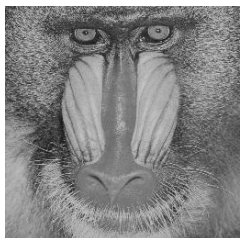
(a)

Citra Lena mewakili citra dengan aktivitas frekuensi rendah,



(b)

Citra paprika mewakili citra dengan aktivitas frekuensi sedang,



(c)

Citra paprika mewakili citra dengan aktivitas frekuensi tinggi.

Gambar 4.1 Citra Inputan

4.3 Kompresi Data

4.3.1 Pembacaan Citra

Pada tahap gambar akan dibaca oleh sistem dan diubah menjadi matrik dengan ukuran (MXN)



90	124	118	119	120	119	118	118
125	172	164	165	167	165	163	164
120	164	157	158	160	158	156	156
119	164	157	158	158	158	156	153
116	162	156	156	158	157	156	154
116	162	156	153	156	156	156	155
117	162	156	154	156	156	155	156
118	163	155	157	156	155	156	155

Gambar 4.2 Pembacaan Citra

Ukuran matrik (MxN) tergantung dari ukuran gambar, sedangkan koefisien tergantung dari intensitas warna dalam gambar, pada tugas akhir ini citra yang digunakan adalah grayscale, sehingga nilai koefisien berkisar antara (0-255).

4.3.2 Sampling

Agar dapat diproses dengan menggunakan komputer, maka suatu citra harus direpresentasikan secara numerik dengan cara *digitalisasi*, yaitu merepresentasikan citra kontinu kedalam bentuk diskrit (digital), baik terhadap koordinat citra, maupun terhadap intensitasnya. Resolusi data chroma diturunkan (downsampling), biasanya dengan faktor pembagian 2 ($256/2 = 128$). Hal ini dikarenakan mata manusia lebih peka terhadap detail brightness daripada detail warna. Misalnya setelah hasil sampling nilai citra sbb :

90	124	118	119	120	119	118	118
125	172	164	165	167	165	163	164
120	164	157	158	160	158	156	156
119	164	157	158	158	158	156	153
116	162	156	156	158	157	156	154
116	162	156	153	156	156	156	155
117	162	156	154	156	156	155	156
118	163	155	157	156	155	156	155

-128 =

165	43	49	43	44	40	39	14
-37	3	0	3	3	1	1	0
-45	2	4	1	3	1	1	0
-49	0	3	1	0	1	2	2
-45	2	0	2	1	3	0	-1
-39	2	2	2	1	2	2	0
-30	1	2	1	1	1	0	1
-16	0	0	0	1	0	1	1

Gambar 4.3 Proses Sampling

Pengurangan dengan nilai 128 bertujuan agar diperoleh nilai diseputar nol.

4.3.3 Discrete Cosine Transform

Proses sebelumnya dilakukan pembagian gambar menjadi blok 8x8. DCT akan diterapkan pada tiap-tiap blok. Untuk koefisien DC dari masing-masing blok akan digunakan pengkodean prediktif dimana nilai dari DC akan diprediksi dengan nilai DC dari blok sebelumnya dan bila terjadi kesalahan maka akan dikuantisasi secara bersama-sama. Sedangkan untuk koefisien AC akan dikuantisasi secara langsung menggunakan quantizer yang berbeda. Kesalahan prediksi nilai dari AC dan DC akan ditentukan dalam normalisasi matrik.

```
[mf,nf]=size(f);
mb=mf/8; nb=nf/8;

Ff = blkproc(f,[8 8],'dct');
Ff = blkproc(Ff',[8 8],'dct');
Ff = round(Ff');
```

4.3.4 Quantization

Proses membersihkan koefisien DCT yang tidak penting untuk pembentukan image baru. Hal ini yang menyebabkan JPEG bersifat lossy. Berikut perintah yang digunakan untuk melakukan kuantisasi :

```
Q = zeros(8);
for i = 1:8,
    for j = 1:8,
        Q(i,j) = 50;
    end
end
```

4.3.5 Entropy Coding

Proses penggunaan algoritma entropy, misalnya Huffman atau RLE untuk mengkodekan koefisien hasil proses DCT yang akan mengeliminasi nilai-nilai matriks yang bernilai nol secara zig-zag order.

Berikut program yang digunakan dalam pengkodean DCT :

```
Fq=round(blkproc(Ff,[8 8],'divq',Q));
if mb*nb > 1,
    fdc=reshape(Fq(1:8:mf,1:8:nf)',mb*nb,1);
    fdpcm=dpcm(fdc,1);
else
    fdpcm=Fq(1,1);
end

dccof=[];
for i=1:mb*nb,
    dccof=[dccof jdcenc(fdpcm(i))];
end

z=[1    2    6    7   15   16   28   29
    3    5    8   14   17   27   30   43
    4    9   13   18   26   31   42   44
   10   12   19   25   32   41   45   54
   11   20   24   33   40   46   53   55
   21   23   34   39   47   52   56   61
   22   35   38   48   51   57   60   62
   36   37   49   50   58   59   63   64];

acseq=[];
for i=1:mb
    for j=1:nb
        tmp(z)=Fq(8*(i-1)+1:8*i,8*(j-1)+1:8*j);
        % tmp is 1 by 64
        eobi=max(find(tmp~=0)); %end of block index
        % eob is labelled with 999
        acseq=[acseq tmp(2:eobi) 999];
    end
end
accf=jacenc(acseq);
```


Sehingga menghasilkan output sebagai berikut :

```

Command Window
DC coefficient after Huffman coding has 4952 bits
AC coefficient after Huffman coding has 26328 bits
Compression Rate    0.47729    Bits / pixel
Compression Ratio    16.7611 : 1
Elapsed time is 1.668635 seconds.
>> |

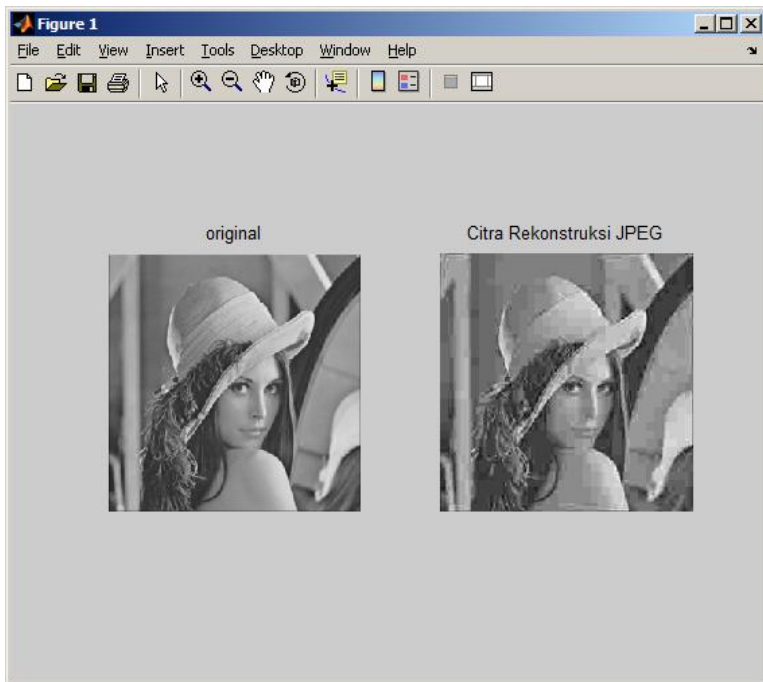
```

Gambar 4.4 Rate Kompresi dan Ratio Kompresi

Dari gambar 4.2 dapat diketahui nilai koefisien DC setelah dikodekan menggunakan huffman coding yang bernilai 4952 bits dan koefisien AC bernilai 26328 bits. Selain mengetahui jumlah bits yang dihasilkan oleh kedua koefisien diatas kita juga mengetahui rate kompresi dari tiap-tiap gambar dan ratio perbandingan kompresi antara gambar original dengan gambar yang telah mengalami rekonstruksi. Dalam simulasi ini dapat diketahui berapa lama dilakukan pengkompresian pada sebuah gambar dengan quality tertentu.

4.3.6 Proses Dekoding

Setelah kuantisasi selesai barulah dilakukan sebuah simulasi pada ketiga citra. Citra yang dipilih akan di proses mulai dari kompresi, pengacakan data yang menggunakan gold code kemudian di transmisikan sehingga dapat diterima oleh receiver. Data yang akan diterima masih berupa bit-bit binary yang nantinya akan di invers menggunakan algoritma DCT. Berikut adalah hasil dari proses kompresi suatu citra lena :



Gambar 4.5 Kompresi Citra Lena

Dapat dilihat bahwa adanya perbedaan antara citra original dengan citra hasil rekonstruksi. Pada citra rekonstruksi terlihat bahwa gambar mengalami rusak namun tidak begitu parah. Rusak atau tidaknya citra tergantung pada level kuantisasi yang diberikan pada citra. Citra lena ini memiliki gradasi yang tinggi. Bila suatu citra memiliki gradasi yang tinggi maka citra tersebut cenderung berada pada frekuensi terendah.

4.4 PSNR

Dalam tugas akhir ini parameter yang akan diukur adalah PSNR. [3] merupakan nilai rata-rata pengkodean dari tiap pixel pada citra. Rate dapat digunakan untuk menghitung besar file dengan cara mengalikan dengan ukuran citra. Sesuai dengan persamaan 2.5 maka dapat dibikin sebuah program seperti berikut :

```
% Calculate MSE , SNR
f = f + 128;
MSE = sum(sum((f-iFf).^2))/(256^2)
PSNR = 10*log10(255^2/MSE)
```

Sehingga akan menghasilkan output sebagai berikut :

```
Command Window
DC coefficient after Huffman coding has 4952 bits
AC coefficient after Huffman coding has 26328 bits
Compression Rate 0.47729 Bits / pixel
Compression Ratio 16.7611 : 1

MSE =

    47.4965

PSNR =

    31.3642

Elapsed time is 3.560309 seconds.
>>
```

Gambar 4.6 MSE dan PSNR dari Citra Lena dengan Q = 50

Terlihat bahwa citra yang telah di kuantisasi sebesar 50 akan menghasilkan PSNR sebesar 31.36. Selanjutnya hasil nilai PSNR dan rasio kompresi dibandingkan dengan mengacu pada tabel 4-1. sehingga dapat ditentukan parameter quality yang tepat untuk mendapatkan citra hasil kompresi JPEG yang baik.

Tabel 4-2 Nilai PSNR

PSNR(dB)	Picture Quality
60	Excellent, no noise apparent
50	Good, a small amount of noise but picture quality good
40	Reasonable, fine grain or snow in the picture, some fine detail lost
30	Poor picture with a great deal of noise
20	Unusable

4.5 Eksperimen pada Citra

Berubahnya nilai kuantisasi sangat berpengaruh pada nilai PSNR dari suatu citra. Pada tugas akhir ini dilakukan pengambilan data serta membandingkan nilai kuantisasi mulai 5 hingga 100 pada tiap-tiap citra. Setiap citra akan menghasilkan nilai PSNR yang berbeda. Selain nilai PSNR, dilakukan pula perbandingan ratio kompresi antara citra asal dengan citra yang telah terekonstruksi.

4.5.1 Citra Lena

Citra lena yang digunakan berukuran 256x 256 bertipe *.bmp. berikut data yang dihasilkan :

Tabel 4-3 Hasil Kompresi JPEG pada Citra Lena

Q	PSNR	Ratio Kompresi	Rate Kompresi (bits/pixel)	MSE	Elapsed Time (Second)
100	45.89	3:1	2.16	118.16	3.7609
95	41.30	5:1	1.49	111.10	3.6721
90	38.81	7:1	1.12	104.70	3.8478
85	36.92	8:1	0.95	97.63	4.0707
80	35.53	10:1	0.79	90.66	4.1933
75	34.42	11:1	0.7	83.47	4.7239
70	33.52	13:1	0.61	76.50	3.999
65	32.72	14:1	0.56	68.73	3.3873
60	32.05	15:1	0.51	61.40	4.688
55	31.36	16:1	0.47	54.52	5.022
50	30.76	18:1	0.43	47.49	5.314
45	30.25	19:1	0.41	40.51	5.344
40	29.75	21:1	0.37	34.74	5.6839
35	29.24	22:1	0.35	28.87	6.4613
30	28.91	24:1	0.32	23.46	6.7223
25	28.55	25:1	0.31	18.19	4.9057
20	28.23	27:1	0.28	13.19	5.5061
15	27.93	28:1	0.27	8.53	9.732
10	27.67	30:1	0.26	4.81	12.661
5	27.40	31:1	0.25	1.67	23.294



Gambar 4.7 Citra Lena dengan $Q = 70$; $\text{bpp} = 0.61$

4.5.2 Citra Baboon

Citra baboon yang digunakan berukuran 256×256 bertipe *.bmp. berikut data yang dihasilkan :

Tabel 4-4 Hasil Kompresi JPEG pada Citra Baboon

Q	PSNR	Ratio Kompresi	Rate Kompresi (bits/pixel)	MSE	Elapsed Time (Second)
100	48.23	2:1	3.63	374.24	5.4897
95	42.57	3:1	2.82	347.97	6.4095
90	38.94	3:1	2.42	320.87	6.9648
85	35.70	4:1	2.16	293.10	7.4627
80	34.29	4:1	1.55	265.89	8.2314
75	32.83	4:1	1.44	236.95	8.2261
70	31.27	5:1	1.38	210.04	9.5485
65	30.07	5:1	1.3	182.28	10.1833
60	29.11	5:1	1.19	156.22	11.0568
55	28.00	6:1	1.18	128.93	11.742
50	27.02	6:1	1.17	102.86	12.320
45	26.19	7:1	1.16	79.66	13.707
40	25.52	7:1	1.07	63.94	16.858
35	24.90	8:1	1.03	48.44	18.394
30	24.38	8:1	0.95	33.81	21.826
25	23.88	9:1	0.91	24.19	22.506
20	23.46	9:1	0.84	17.48	19.624
15	23.06	10:1	0.8	8.29	23.475
10	22.71	10:1	0.75	3.59	32.828
5	23.39	11:1	0.7	0.97	38.768



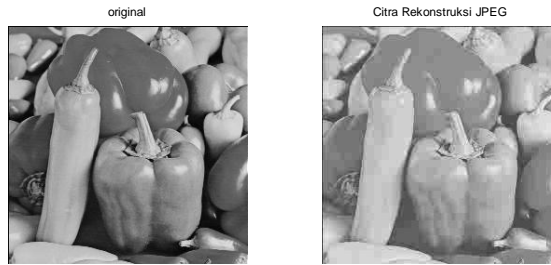
Gambar 4.8 Citra Baboon dengan $Q = 70$; $\text{bpp} = 1.38$

4.5.3 Citra Pepper

Citra pepper yang digunakan berukuran 256×256 bertipe *.jpg . Berikut data yang dihasilkan :

Tabel 4-5 Hasil Kompresi JPEG pada Citra Pepper

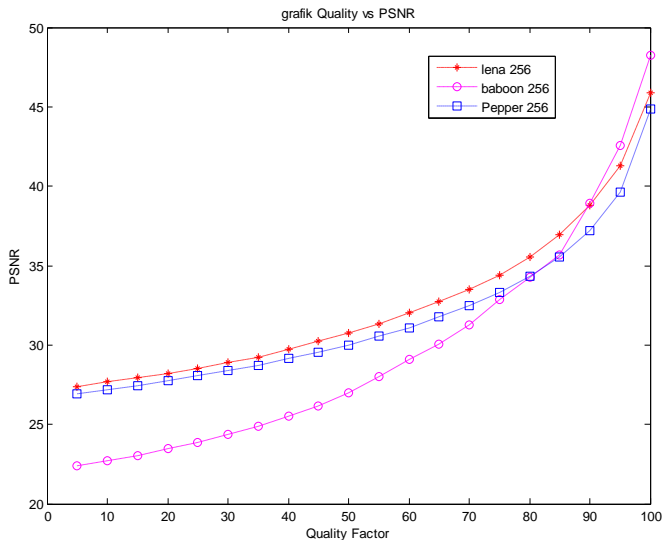
Q	PSNR	Ratio Kompresi	Rate Kompresi (bits/pixel)	MSE	Elapser Time (Second)
100	44.87	3 :1	2.96	132.19	3.9193
95	39.62	4:1	1.99	125.27	3.964
90	37.23	6:1	1.34	117.18	4.0376
85	35.55	7:1	1.09	109.53	4.223
80	34.34	9:1	0.89	101.79	4.461
75	33.32	10:1	0.79	94.31	4.5334
70	32.47	11:1	0.69	86.71	4.6439
65	31.75	12:1	0.63	79.42	4.6779
60	31.08	14:1	0.57	72.15	4.8196
55	30.56	15:1	0.53	64.83	5.1103
50	30.01	16:1	0.48	57.07	5.907
45	29.54	17:1	0.46	50.60	6.2136
40	29.13	18:1	0.42	43.41	6.5334
35	28.74	19:1	0.4	36.75	7.4645
30	28.38	21:1	0.37	30.26	8.3864
25	28.05	22:1	0.35	23.92	8.7944
20	27.73	23:1	0.34	18.07	10.57
15	27.44	24:1	0.32	12.28	14.308
10	27.15	25:1	0.31	7.09	24.123
5	26.91	26:1	0.3	2.11	39.526



Gambar 4.9 Citra Pepper dengan $Q = 70$; $\text{bpp} = 0.69$

4.6 Pengaruh Quality

Suatu citra dapat mengalami kerenggangan frekuensi dikarenakan berubahnya nilai kualitas (Q). Dengan semakin besarnya nilai Q maka kualitas gambar akan semakin kabur atau tidak jelas. Dan nilai dari Q juga berpengaruh pada nilai PSNR dari citra yang diproses. Semakin besar nilai Q maka nilai PSNR dari citra akan semakin menurun. Berikut grafik yang di dapat :

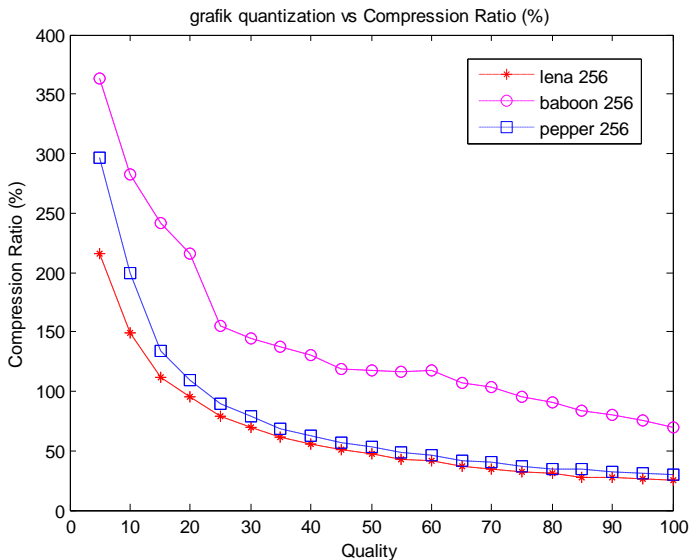


Gambar 4.10 Grafik hubungan antara quality dan nilai PSNR

Terlihat pada gambar 4.9 bahwa faktor quality mempengaruhi nilai PSNR dari suatu citra. Dari ketiga jenis citra yaitu lena, baboon, dan pepper terlihat bahwa citra baboon lah yang memiliki nilai PSNR lebih rendah dari pada lena dan pepper. Dengan melihat tabel 4-1 maka citra baboon tergolong citra yang kualitas citranya rendah (buruk).

4.7 Pengaruh Rasio Kompresi

Selain pengaruh kualitas terhadap nilai PSNR, adapula pengaruh dari quality terhadap besar kompresi citra. Nilai kompresi suatu citra merupakan hasil perbandingan antara citra asal dengan citra yang telah mengalami kompresi. Berikut grafik yang terjadi antara Quality (Q) dengan Rasio Kompresi dari ketiga citra yaitu lena, baboon, dan pepper :



Gambar 4.11 Grafik hubungan antara Quality dengan Compression Ratio


Pada gambar 4.10 terlihat adanya hubungan antara Quality dengan Compression Ratio (%) pada masing- masing citra. Dari ketiga citra diatas dapat dilihat bahwa untuk citra lena ratio kompresi terbesar

sekitar 216, baboon 363 dan pepper 296. Dapat diambil keputusan bahwa semakin besar faktor kualitas citra maka akan semakin besar pula ratio kompresi dari suatu citra. Namun ratio kompresi yang baik bila memiliki nilai ratio yang kecil.

4.8 Perbandingan Citra

Citra asal dengan citra setelah rekonstruksi memiliki perubahan nilai dari warna citra. Perubahan yang terjadi diakibatkan terkenanya kompresi pada citra yang akan dikirim. Berikut ini adalah perubahan yang terjadi pada citra asal :

90	124	118	119	120	119	118	118
125	172	164	165	167	165	163	164
120	164	157	158	160	158	156	156
119	164	157	158	158	158	156	153
116	162	156	156	158	157	156	154
116	162	156	153	156	156	156	155
117	162	156	154	156	156	155	158
118	163	155	157	156	155	156	155



153	153	153	153	153	153	153	153
153	153	153	153	153	153	153	153
153	153	153	153	153	153	153	153
153	153	153	153	153	153	153	153
153	153	153	153	153	153	153	153
153	153	153	153	153	153	153	153
153	153	153	153	153	153	153	153
153	153	153	153	153	153	153	153

Gambar 4.12 Perbandingan Citra Asal dengan Citra Rekonstruksi

BAB V

PENUTUP

5.1 KESIMPULAN

Berdasarkan hasil analisa dan pembahasan, diperoleh kesimpulan sebagai berikut :

1. Kompresi JPEG yang disimulasikan dengan perangkat lunak menggunakan tiga citra digital ternyata dapat memberikan hasil berupa kualitas citra yang baik dan rasio kompresi yang tinggi jika digunakan parameter quality di atas 75. Dimana pada parameter ini diperoleh nilai PSNR antara 32,83 sampai 48,23 dan jumlah bit per pixel 3.63 bpp.
2. Faktor kualitas citra tertinggi berada pada angka 100 menunjukan bahwa citra tersebut tergolong excellent namun bila faktor kualitas suatu citra dibawah angka 75 maka citra tersebut tergolong poor / buruk

5.2 SARAN

Mengingat masih banyaknya perbaikan yang perlu dilakukan pada proyek akhir ini, maka penulis mempertimbangkan beberapa saran yang diperlukan dalam proses perbaikan-perbaikan pada proyek akhir ini diantaranya adalah:

1. Melakukan pengujian terhadap citra berwarna
2. Melakukan penransmisian spread spektrum dengan sinkronisasi
3. Melakukan perbandingan antara kompresi JPEG dengan kompresi berbasis fractal atau wavelet

=Halaman ini sengaja dikosongkan=

DAFTAR PUSTAKA

- [1] Bandemer Bernd, 2003, Course Project 4 ECE 642,
<http://www.stud.tuilmennau.de/~bebaii/docs.html>
- [2] Krikor Lala, Sami Baba, Thawar Arif, “*Image Encryption Using DCT and Stream Cipher*”, European Journal of Scientific Research, ISSN 1450-216X Vol.32 No.1 (2009), pp.47-57
- [3] Li, Drew. “Image Compression Standart”, Fundamental of Multimedia, Chapter 9, *Prentice Hall*, 2003.
- [4] M. I. Khalil “*Image Compression Using New Entropy Coder*”, Paper On international Journal of Computer Theory and Engineering, Vol 2, february 2010
- [5] Lin. Y.T, F. H. Yeh, and G. C. Lee, “*Removable Visible Watermarking in JPEG Compression Domain*”, paper on IEEE Asia-Pacific Services Computing Conference, 2008
- [6] R Munir, 2004, “Pengolahan Citra Digital dengan pendekatan algoritmik”, Penerbit Informatika, Bandung
- [7] Shah, Punnet, “*Gold Code Sequence*”, National Institute of Technology, Durgapur 2008
- [8] W.Y. Wang, “*New Lossless Compression Method for JPEG*”, Pacific-Asia Conference on Circuits, Communications and System 2009 Datasheet SRF04.
- [9] Winanti Winda, “*Penyembunyian Pesan Citra Terkompresi JPEG*” menggunakan Metode Spread Spektrume, Institut Teknologi Bandung, 2004. Datasheet PIR35.

=Halaman ini sengaja dikosongkan=

LAMPIRAN

➤ Citra Lena

Nilai $Q = 10$; PSNR 27.40

original



Citra Rekonstruksi JPEG



Nilai $Q = 50$; PSNR 31.36

original

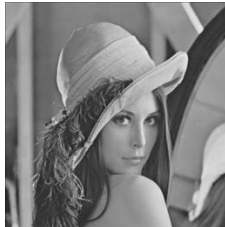


Citra Rekonstruksi JPEG



Nilai $Q = 100$; PSNR 41.30

original



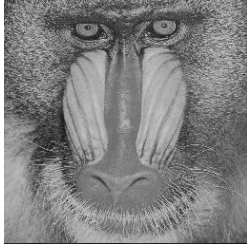
Citra Rekonstruksi JPEG



➤ Citra Baboon

Nilai $Q = 10$; PSNR 22.39

original

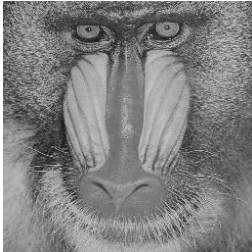


Citra Rekonstruksi JPEG

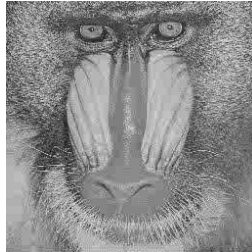


Nilai $Q = 50$; PSNR 28.00

original

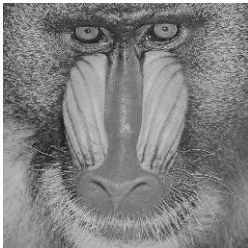


Citra Rekonstruksi JPEG

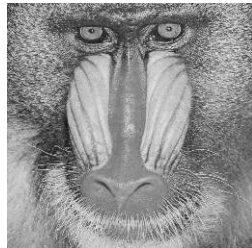


Nilai $Q = 100$; PSNR 42.57

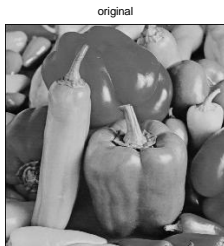
original



Citra Rekonstruksi JPEG



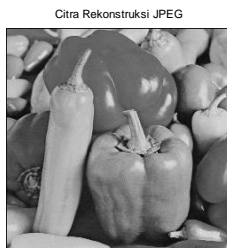
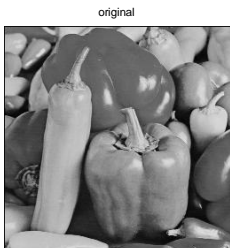
- Citra Pepper
Nilai Q = 10 ; PSNR 26.91



Nilai Q = 50 ; PSNR 30.56



Nilai Q = 100 ; PSNR 39.62



➤ JPEG.m

```

clear all, close all, clc;
tic

%----- Baca Citra -----
I = imread('pep.jpg');
f = I;
subplot 121
imshow(I),title ('original')

%----- DCT -----
f = double(f);
f = f - 128;

[mf,nf]=size(f);
mb=mf/8; nb=nf/8;

Ff = blkproc(f,[8 8],'dct');
Ff = blkproc(Ff',[8 8],'dct');
Ff = round(Ff');

Q = zeros(8);
for i = 1:8,
    for j = 1:8,
        Q(i,j) = 50;
    end
end

Fq=round(blkproc(Ff,[8 8],'divq',Q));

if mb*nb > 1,
    fdc=reshape(Fq(1:8:mf,1:8:nf)',mb*nb,1);
    fdpcm=dpcm(fdc,1);
else
    fdpcm=Fq(1,1);
end

dccof=[];
for i=1:mb*nb,
    dccof=[dccof jdcenc(fdpcm(i))];
end

z=[1 2 6 7 15 16 28 29
```

```

    3    5    8   14   17   27   30   43
    4    9   13   18   26   31   42   44
   10   12   19   25   32   41   45   54
   11   20   24   33   40   46   53   55
   21   23   34   39   47   52   56   61
   22   35   38   48   51   57   60   62
   36   37   49   50   58   59   63   64];

acseq=[];
for i=1:mb
    for j=1:nb
        tmp(z)=Fq(8*(i-1)+1:8*i,8*(j-1)+1:8*j);
        % tmp is 1 by 64
        eobi=max(find(tmp~=0)); %end of block index
        % eob is labelled with 999
        acseq=[acseq tmp(2:eobi) 999];
    end
end
accf=jacenc(acseq);

disp(['DC coefficient after Huffman coding has '
int2str(length(dccof)) ...
' bits'])
disp(['AC coefficient after Huffman coding has '
int2str(length(accf)) ...
' bits'])

disp(['Compression Rate '
num2str((length(dccof)+length(accf))/(mb*nb*64)) '
Bits / pixel '])
disp(['Compression Ratio '
num2str(8/((length(dccof)+length(accf))/(mb*nb*64)))
' : 1'])

%----gold code----
Xa=[1 1 0 0 1 1 1 1];
Xb=[1 0 0 0 0 1 1 1];
for w=1:255
    Xa=[mod(Xa(8)+Xa(7)+Xa(6)+Xa(5)+Xa(2)+Xa(1),2)
Xa(1:7)];
    seq1(w)=Xa(8);
    Xb=[mod(Xb(8)+Xb(7)+Xb(6)+Xb(1),2) Xb(1:7)];
    seq2(w)=Xb(8);
end

```

```

        goldcode(w)=mod(Xa(8)+Xb(8),2);
    end
    for m=2:255
        goldcode(m,:)=wshift('1D',goldcode(m-1,:),1);
    end
    gld=reshape(goldcode,1,[]);

%----kanal BSC-----
y1=[accof(1:end)];
temp=10^(-3);
for v=1:length(y1)
    r=rand(1);
    if (r<temp);
        y1(v)=(accof(v));
    end
end

y2=[dccof(1:end)];
temp=10^(-3);
for s=1:length(y2)
    r=rand(1);
    if (r<temp);
        y2(s)=(dccof(s));
    end
end

[N,BER]=biterr(accof,y1);
[N,BER]=biterr(dccof,y2);

%----- IDCT -----
acarr=jacdec(y1);
dcarr=jdcdec(y2);

z=z(:);

Eob=find(acarr==999);
kk=1;ind1=1;n=1;
for ii=1:mb
    for jj=1:nb
        ac=acarr(ind1:Eob(n)-1);
        ind1=Eob(n)+1;
        n=n+1;
        ri(8*(ii-1)+1:8*ii,8*(jj-
1)+1:8*jj)=dezz([dcarr(kk) ac zeros(1,63-
length(ac))]);
    end
end

```

```

        kk=kk+1;
    end
end

iFq=round(blkproc(ri,[8 8],'idivq',Q));
iFf=blkproc(iFq,[8 8],'idct2');
iFf=round(iFf+128);
subplot 122
imshow(mat2gray(iFf)),title('Citra Rekonstruksi JPEG')

% Calculate MSE , SNR
f = f + 128;
MSE = sum(sum((f-iFf).^2))/(256^2)
PSNR = 10*log10(255^2/MSE)

```

Toc

➤ **divq.m**

```

function y=divq(x,q)
% Usage: y=divq(x,q)
% element by element matrix division with known
quotient matrix
% x and q must have the same size
y =x./q;

```

➤ **dpcm.m**

```

function [r,xtilde]=dpcm(x,a)

% Differential Pulse Coded Modulation
% x: vektor masukan yang akan dikodekan
% a: prediction filter
% r: vektor sisa dikuantisasi ke integer

[m,nx]=size(x);

if m==1 & nx > 1, x = x'; m=nx; end % Memastikan
b bahwa x merupakan vektor kolom
[p,na]=size(a);
if p==1 & na > 1, a = a'; p=na; end % Memastikan
b bahwa a merupakan vektor kolom

% sejumlah p elemen pertama dari x harus dikirim tanpa
diubah sebagai bagian r
% setelah kuantisasi skalar

```

```

r=round(x(1:p)); xtilde=r;
for t=p+1:m,
    xhat(t)=a'*xtilde(t-1:-1:t-p);
    r(t)=round(x(t)-xhat(t));
    xtilde(t)=xhat(t)+r(t);
end

```

➤ **jdcenc.m**

```

function b=jdcenc(x)
% Usage: b=jdcenc(x)
% Hoffman encoding of DC coefficients in JPEG

% category c = floor(log2(abs(x)))+1
% append code is the binary representation of abs(x)
if x>0
% of the 1's complement of binary rep of abs(x) if x <
0
% first figure out category number
if x ==0,
    %b=[0 1 0];
    b=[0 0];
    return % done
else
    c = floor(log2(abs(x)))+1;
end

```

```

% Hoffman table
tbl=[3 0 1 0 0 0 0 0 0 0
     3 0 1 1 0 0 0 0 0 0
     3 1 0 0 0 0 0 0 0 0
     2 0 0 0 0 0 0 0 0 0
     3 1 0 1 0 0 0 0 0 0
     5 1 1 0 0 0 0 0 0 0
     4 1 1 1 0 0 0 0 0 0
     5 1 1 1 1 0 0 0 0 0
     6 1 1 1 1 1 0 0 0 0
     7 1 1 1 1 1 1 0 0 0
     8 1 1 1 1 1 1 1 0 0
     9 1 1 1 1 1 1 1 1 0];

```

% This table is in the pdf file, print out taken
 % Modified 27 sept 2003, 2325 Hrs

```

tab=[2 0 0 0 0 0 0 0 0 0

```

```

3 0 1 0 0 0 0 0 0 0 0
3 0 1 1 0 0 0 0 0 0 0
3 1 0 0 0 0 0 0 0 0 0
3 1 0 1 0 0 0 0 0 0 0
3 1 1 0 0 0 0 0 0 0 0
4 1 1 1 0 0 0 0 0 0 0
5 1 1 1 1 0 0 0 0 0 0
6 1 1 1 1 1 0 0 0 0 0
7 1 1 1 1 1 1 0 0 0 0
8 1 1 1 1 1 1 1 0 0 0
9 1 1 1 1 1 1 1 1 0 0];

tbl=tab;
% This is changed by Ravi Lakkundi

b=tbl(c+1,2:tbl(c+1,1)+1);

tmp=int2bin(x,c);
% tmp is 1 by c+1 vector containing sign-mag
% representation of x, first bit is sign bit.

if tmp(1)==0, % if x > 0
    b=[b tmp(2:c+1)];
elseif tmp(1)==1, % if x < 0
    b=[b ones(1,c)-tmp(2:c+1)];
end

➤ jacenc.m
function b=jacenc(x)
% Usage: b=jacenc(x)
% JPEG AC coefficient encoding routine
% x: AC coefficient sequence
% b: corresponding bit stream

b=[];
% run - category - length - bsae code length -
base code
table=[...
0 1 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 2 4 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 6 3 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 4 8 4 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 5 10 5 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 6 12 6 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

```

```

0 7 14 7 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 8 18 10 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0
0 9 25 16 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 0
0 10 26 16 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1
1 1 5 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 2 8 6 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0
1 3 10 7 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0
1 4 13 9 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0
1 5 16 11 1 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0
1 6 22 16 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0
1 7 23 16 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1
1 8 24 16 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0
1 9 25 16 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1
1 10 26 16 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 0
2 1 6 5 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0
2 2 10 8 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
2 3 13 10 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0
2 4 20 16 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 1
2 5 21 16 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 0
2 6 22 16 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1
2 7 23 16 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0
2 8 24 16 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 1
2 9 25 16 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 0
2 10 26 16 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1
3 1 7 6 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0
3 2 11 9 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0
3 3 14 11 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0
3 4 20 16 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0
3 5 21 16 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 1
3 6 22 16 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0
3 7 23 16 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1
3 8 24 16 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 0
3 9 25 16 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1
3 10 26 16 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 0
4 1 7 6 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0
4 2 12 10 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
4 3 19 16 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1
4 4 20 16 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 0
4 5 21 16 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 1
4 6 22 16 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 0
4 7 23 16 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 1
4 8 24 16 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 0
4 9 25 16 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1
4 10 26 16 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0

```

```

5 1 8 7 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0
5 2 12 10 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0
5 3 19 16 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1
5 4 20 16 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0
5 5 21 16 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 1
5 6 22 16 1 1 1 1 1 1 1 1 1 0 1 0 0 0 1 0
5 7 23 16 1 1 1 1 1 1 1 1 1 0 1 0 0 0 1 1
5 8 24 16 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 0
5 9 25 16 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1
5 10 26 16 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 0
6 1 8 7 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0
6 2 13 11 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
6 3 19 16 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1
6 4 20 16 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 0
6 5 21 16 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 1
6 6 22 16 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0
6 7 23 16 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1
6 8 24 16 1 1 1 1 1 1 1 1 1 0 1 0 1 1 0 0
6 9 25 16 1 1 1 1 1 1 1 1 1 0 1 0 1 1 0 1
6 10 26 16 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0
7 1 9 8 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0
7 2 13 11 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0
7 3 19 16 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1
7 4 20 16 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 0
7 5 21 16 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 1
7 6 22 16 1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 0
7 7 23 16 1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1
7 8 24 16 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 0
7 9 25 16 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1
7 10 26 16 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 0
8 1 9 8 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0
8 2 17 15 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0
8 3 19 16 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1
8 4 20 16 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0
8 5 21 16 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1
8 6 22 16 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0
8 7 23 16 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1
8 8 24 16 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0
8 9 25 16 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1
8 10 26 16 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0
9 1 10 9 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
9 2 18 16 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
9 3 19 16 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
9 4 20 16 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1

```


9	5	21	16	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0
9	6	22	16	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
9	7	23	16	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0
9	8	24	16	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1
9	9	25	16	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0
9	10	26	16	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1
10	1	10	9	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0
10	2	18	16	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0
10	3	19	16	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	1
10	4	20	16	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0
10	5	21	16	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
10	6	22	16	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0
10	7	23	16	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1
10	8	24	16	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
10	9	25	16	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
10	10	26	16	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0
11	1	10	9	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
11	2	18	16	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1
11	3	19	16	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0
11	4	20	16	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1
11	5	21	16	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0
11	6	22	16	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1
11	7	23	16	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0
11	8	24	16	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1
11	9	25	16	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0
11	10	26	16	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1
12	1	11	10	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0
12	2	18	16	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0
12	3	19	16	1	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1
12	4	20	16	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0
12	5	21	16	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
12	6	22	16	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0
12	7	23	16	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
12	8	24	16	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
12	9	25	16	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
12	10	26	16	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0
13	1	12	11	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0
13	2	18	16	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1
13	3	19	16	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0
13	4	20	16	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
13	5	21	16	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0
13	6	22	16	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
13	7	23	16	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0
13	8	24	16	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1

```

13  9  25 16 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0
13 10  26 16 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1
14  1  13 12 1 1 1 1 1 1 1 1 0 1 1 0 0 0 0 0
14  2  18 16 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0
14  3  19 16 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1
14  4  20 16 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0
14  5  21 16 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
14  6  22 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
14  7  23 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1
14  8  24 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0
14  9  25 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1
14 10  26 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0
15  1  17 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1
15  2  18 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0
15  3  19 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
15  4  20 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
15  5  21 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1
15  6  22 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0
15  7  23 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
15  8  24 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
15  9  25 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
15 10  26 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
  0  0   4  4 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
15  0  12 12 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0];

ix = find(x~=0); % index of next non-zero entry
Nx = length(ix); % # of non-zero entries

% if first ac coefficient has no leading zeros
% then encode x(1)!
% note much details have been left out here
% we did not concern what if Nx=1 and x(1)=EoB

if ix(1)==1,
    run=0; % no leading zero
    tmp=int2bin(x(1)); % tmp is 1 by (c+1)
    cat=length(tmp)-1; % first bit is sign bit
    row=run*10+cat;
    prefix=table(row,5:table(row,4)+4);
    if tmp(1)==1, %if current x is negative
        tmp=ones(1,cat+1)-tmp; % 1's complement
    end
    b=[prefix tmp(2:cat+1)];
end

```

```

for n=2:Nx, % loop through each non-zero entry
    prefix=[];
    % 1. ck # of leading zeros
    run=ix(n)-ix(n-1)-1;
    % 2. ck if run > 16;
    % run of 0s longer than 15, will be broken down
    into repeated
    % run-of-15-0 symbol
    while run > 15,
        b=[b 1 1 1 1 1 1 1 1 0 1 1 1];
        run=run-15;
    end
    % check if end of block
    if x(ix(n))==999,
        b = [b 1 0 1 0]; % append EoB symbol
    else
        tmp=int2bin(x(ix(n))); % tmp is 1 by (c+1)
        cat=length(tmp)-1; % first bit is sign bit
        row=run*10+cat; % row number of the prefix
        Hoffman code
        prefix=table(row,5:table(row,4)+4);
        if tmp(1)==1, % if current x is negative
            tmp=ones(1,cat+1)-tmp; % 1's complement
        end
        b=[b prefix tmp(2:cat+1)];
    end
end
end

```

➤ **jacdec.m**

```

function x=jacdec(y)
% run - category - length - bsae code length -
base code
table=[...
0 1 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 2 4 2 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 6 3 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 4 8 4 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 5 10 5 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 6 12 6 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 7 14 7 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 8 18 10 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0
0 9 25 16 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0
0 10 26 16 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1

```

```

1 1 5 4 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 2 8 6 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0
1 3 10 7 1 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0
1 4 13 9 1 1 1 1 1 0 1 1 0 0 0 0 0 0 0 0
1 5 16 11 1 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0
1 6 22 16 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0
1 7 23 16 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1
1 8 24 16 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0
1 9 25 16 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1
1 10 26 16 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0
2 1 6 5 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
2 2 10 8 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
2 3 13 10 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0
2 4 20 16 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 1
2 5 21 16 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1 0
2 6 22 16 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1 1
2 7 23 16 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0
2 8 24 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 1
2 9 25 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0
2 10 26 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1
3 1 7 6 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 2 11 9 1 1 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0
3 3 14 11 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0
3 4 20 16 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0
3 5 21 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1
3 6 22 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 1 0
3 7 23 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 1 1
3 8 24 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1 0 0
3 9 25 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1 0 1
3 10 26 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1 1 0
4 1 7 6 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
4 2 12 10 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
4 3 19 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1 1 1
4 4 20 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0
4 5 21 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1
4 6 22 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 1 0
4 7 23 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 1 1
4 8 24 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 0
4 9 25 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 1
4 10 26 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 0
5 1 8 7 1 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
5 2 12 10 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0
5 3 19 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1
5 4 20 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

```

5	5	21	16	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	1
5	6	22	16	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	0
5	7	23	16	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1	1
5	8	24	16	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0	0
5	9	25	16	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0	1
5	10	26	16	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	0
6	1	8	7	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0
6	2	13	11	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
6	3	19	16	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1
6	4	20	16	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	0
6	5	21	16	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1
6	6	22	16	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0
6	7	23	16	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1
6	8	24	16	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	0
6	9	25	16	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0	1
6	10	26	16	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0
7	1	9	8	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0
7	2	13	11	1	1	1	1	1	1	1	1	0	0	1	0	0	0	0	0
7	3	19	16	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1
7	4	20	16	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0
7	5	21	16	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	1
7	6	22	16	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	0
7	7	23	16	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1	1
7	8	24	16	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	0
7	9	25	16	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0	1
7	10	26	16	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	0
8	1	9	8	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0
8	2	17	15	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
8	3	19	16	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1	1
8	4	20	16	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0
8	5	21	16	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0	1
8	6	22	16	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0
8	7	23	16	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1
8	8	24	16	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0
8	9	25	16	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1
8	10	26	16	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0
9	1	10	9	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
9	2	18	16	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
9	3	19	16	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
9	4	20	16	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
9	5	21	16	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0
9	6	22	16	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
9	7	23	16	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0
9	8	24	16	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1

9	9	25	16	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0
9	10	26	16	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1
10	1	10	9	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0
10	2	18	16	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0
10	3	19	16	1	1	1	1	1	1	1	1	1	0	0	1	0	0	1
10	4	20	16	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0
10	5	21	16	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1
10	6	22	16	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0
10	7	23	16	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1
10	8	24	16	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0
10	9	25	16	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
10	10	26	16	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0
11	1	10	9	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0
11	2	18	16	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1
11	3	19	16	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0
11	4	20	16	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1
11	5	21	16	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0
11	6	22	16	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1
11	7	23	16	1	1	1	1	1	1	1	1	1	0	1	0	1	1	0
11	8	24	16	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1
11	9	25	16	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0
11	10	26	16	1	1	1	1	1	1	1	1	1	0	1	1	0	0	1
12	1	11	10	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0
12	2	18	16	1	1	1	1	1	1	1	1	1	0	1	1	0	1	0
12	3	19	16	1	1	1	1	1	1	1	1	1	0	1	1	0	1	1
12	4	20	16	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0
12	5	21	16	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1
12	6	22	16	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0
12	7	23	16	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
12	8	24	16	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
12	9	25	16	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1
12	10	26	16	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0
13	1	12	11	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0
13	2	18	16	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
13	3	19	16	1	1	1	1	1	1	1	1	1	0	0	1	0	0	0
13	4	20	16	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0
13	5	21	16	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0
13	6	22	16	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
13	7	23	16	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0
13	8	24	16	1	1	1	1	1	1	1	1	1	0	1	0	0	0	1
13	9	25	16	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0
13	10	26	16	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1
14	1	13	12	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0
14	2	18	16	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0

```

14 3 19 16 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1
14 4 20 16 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0
14 5 21 16 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
14 6 22 16 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0
14 7 23 16 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1
14 8 24 16 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0
14 9 25 16 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1
14 10 26 16 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0
15 1 17 16 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1
15 2 18 16 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0
15 3 19 16 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
15 4 20 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0
15 5 21 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1
15 6 22 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0
15 7 23 16 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
15 8 24 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
15 9 25 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
15 10 26 16 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
0 0 4 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
15 0 12 12 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0];

%y=[0 -122 -253 -1 zeros(1,18) 0 0 0 -1 9 ]
%%y=tobdel(y);
% Start decoding, first decode the prefix then from
that get run,cat,code etc..
% Use the same LOGIC as that of dc decoding
N=length(y);
[p,dm1]=size(table);
x=[];
i=1; d=5;tmp=ones(p,1);
tep=i;
while i<=N,
    % match y(i) to that of the d-th bit in the table
    tmp=tmp.*(table(:,d)==y(i)); % tmp is a vector of 0
and 1 with 1 indicate a match
    if sum(tmp)==1, % narrow down to one symbol, find
it
        d=5; % reset pointer to columns of
table.
        row=find(tmp); % Index of non zero i.e 1
        run=table(row,1);
        cat=table(row,2);
        i=tep+table(row,4)-1;
        tmp=ones(p,1); % Preset temp vector

```

```

        if row==161,num=999;
        else
            num=bin2int(y(i+1:i+cat));
            if (~(num>=2^(cat-1) & num<2^cat)),num=-
1*bin2int(ones(1,cat)-y(i+1:i+cat));end
            end
            i=i+cat;           % Increment to next prefix
            x=[x zeros(1,run) num];
            tep=i+1;
        else
            d=d+1;
        end
        i=i+1;
    end
    %x

% Take a test vector

➤ jdcdec.m
function x=jdcdec(y)
%clear;clc
%y=[jdcenc(-2) jdcenc(0) jdcenc(0) jdcenc(0) jdcenc(-
124)];
% table specified
table=[2 0 0 0 0 0 0 0 0 0 0
        3 0 1 0 0 0 0 0 0 0 0
        3 0 1 1 0 0 0 0 0 0 0
        3 1 0 0 0 0 0 0 0 0 0
        3 1 0 1 0 0 0 0 0 0 0
        3 1 1 0 0 0 0 0 0 0 0
        4 1 1 1 0 0 0 0 0 0 0
        5 1 1 1 1 0 0 0 0 0 0
        6 1 1 1 1 1 0 0 0 0 0
        7 1 1 1 1 1 1 0 0 0 0
        8 1 1 1 1 1 1 1 0 0 0
        9 1 1 1 1 1 1 1 1 0 0];

N=length(y);
[p,dm1]=size(table);
x=[];
x1=[];
i=1; d=2;tmp=ones(p,1);
while i<=N,
    % match y(i) to that of the d-th bit in the table

```



```

    tmp=tmp.*[table(:,d)==y(i)]; % tmp is a vector of 0
and 1 with 1 indicate a match
    if sum(tmp)==1, % narrow down to one symbol, find
it
        d=2; % reset pointer to columns of
table.
        kkt=0;
        for kk=1:length(tmp)
            if(tmp(kk)==1),kkt=kk;end
        end
        cat=kkt-1;
        tmp=ones(p,1);
        if cat==11,i=i+1;end % Because the comparison
ends in last but one column,but still a 0 is left
        x1=y(i+1:i+cat);
        % Check Range
        if(cat ~=0)
            x2=bin2int(x1);
            if(x2>=2^(cat-1) & x2<2^cat)
                x2=x2;
            else
                x1=ones(1,cat)-x1;
                x2=-1*bin2int(x1);
            end
        else
            x2=0;
        end
        % Update decoded vector
        x=[x x2];
        i=i+cat;
    else
        d=d+1;
    end
    i=i+1;
end
x=filter(1,[1 -1],x); % Inverse of DPCM

```

➤ **dezz.m**

```

function x = dezz(z)
% function x = dezz(z)
% undo zz.m (<z> is a vector, <x> a 8x8 block)
zi=[1   2   6   7  15  16  28  29
     3   5   8  14  17  27  30  43
     4   9  13  18  26  31  42  44

```

```

10 12 19 25 32 41 45 54
11 20 24 33 40 46 53 55
21 23 34 39 47 52 56 61
22 35 38 48 51 57 60 62
36 37 49 50 58 59 63 64];% work on columns

% zi = [1 2 6 7
%        3 5 8 13
%        4 9 12 14
%        10 11 15 16];

% zi = [1 2
%        3 4];

% zi=[ 1 2 6 7 15 16 28 29 45 46
66 67 91 92 120 121
%      3 5 8 14 17 27 30 44 47 65
68 90 93 119 122 151
%      4 9 13 18 26 31 43 48 64 69
89 94 118 123 150 152
%      10 12 19 25 32 42 49 63 70 88
95 117 124 149 153 178
%      11 20 24 33 41 50 62 71 87 96
116 125 148 154 177 179
%      21 23 34 40 51 61 72 86 97 115
126 147 155 176 180 201
%      22 35 39 52 60 73 85 98 114 127
146 156 175 181 200 202
%      36 38 53 59 74 84 99 113 128 145
157 174 182 199 203 220
%      37 54 58 75 83 100 112 129 144 158
173 183 198 204 219 221
%      55 57 76 82 101 111 130 143 159 172
184 197 205 218 222 235
%      56 77 81 102 110 131 142 160 171 185
196 206 217 223 234 236
%      78 80 103 109 132 141 161 170 186 195
207 216 224 233 237 246
%      79 104 108 133 140 162 169 187 194 208
215 225 232 238 245 247
%      105 107 134 139 163 168 188 193 209 214
226 231 239 244 248 253
%      106 135 138 164 167 189 192 210 213 227
230 240 243 249 252 254

```

```
%      136  137  165  166  190  191  211  212  228  229
241  242  250  251  255  256];
```

```
zi = zi(:);
x = reshape(z(zi),8,8);
% x = reshape(z(zi),2,2);
%x = reshape(z(zi),4,4);
% x = reshape(z(zi),16,16);
```

```
➤ Idivq.m
function x=idivq(x,q)
x =x.*q;
```